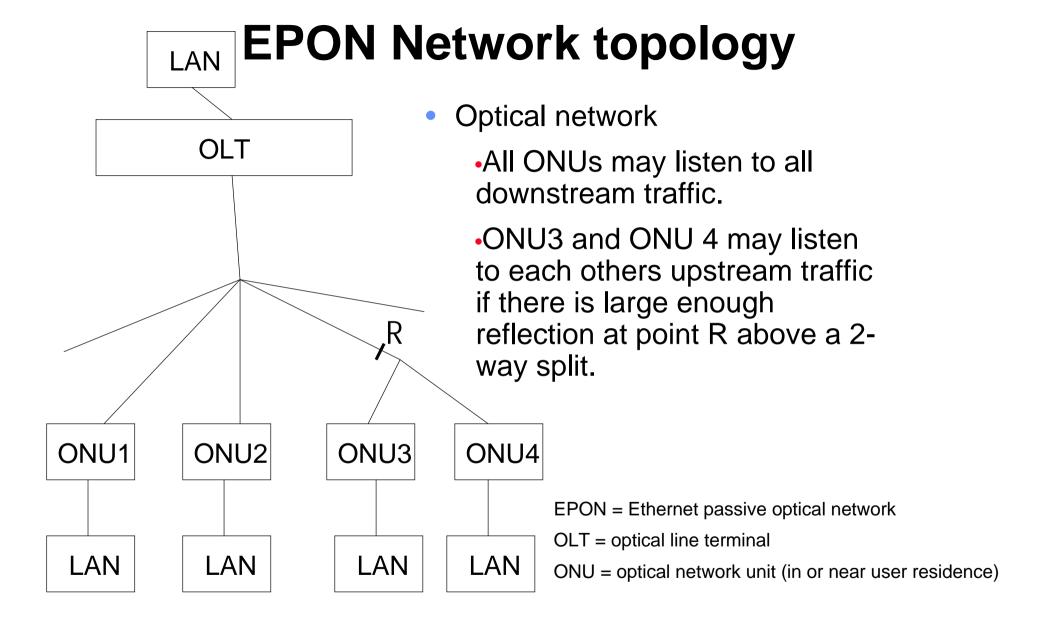
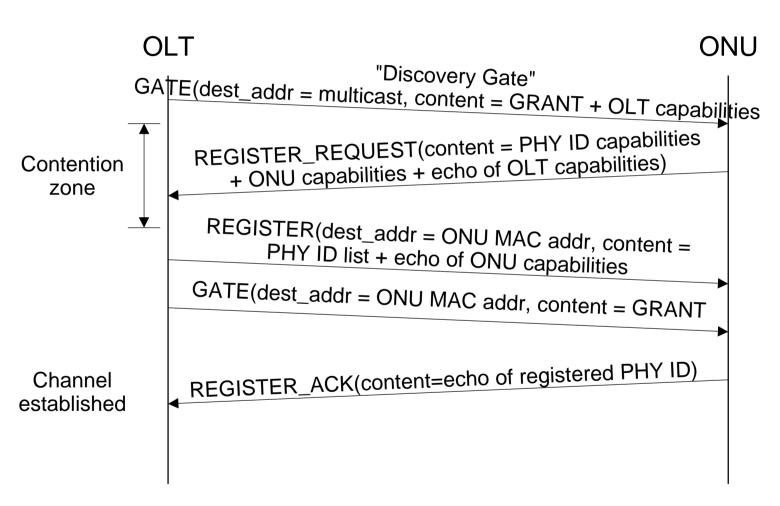
## Key exchange during autodiscovery

Antti Pietiläinen, Nokia



## ONU auto-discovery as of May 2002



Ref. Gaglianello\_1\_0502.pdf

## **Motivation**

- •Confidentiality and privacy over the link are most important. Authentication could be carried out after the encrypted channel is present.
- •Very little sensitive information is revealed if encryption is started immediately -> key exchange should be made part of registration process.
- •If keys are random numbers that do not carry authentication information, they can be generated below MAC in EPON chip hardware. The solution would be internal to 802.3 EPON and would not require interplay between 802 standards. External cryptographic standards could be used.
- Keys would not have to be passed across network layers.
- •802.1x could be used for authentication after the encrypted link is up. The 802.1x protocol could be carried out without special requirements from EPON.

## ONU auto-discovery w. key exchange and encryption. 802.1x port authentication protocol is used for authentication.

**OLT** ONU "Discovery Gate" GATE(dest\_addr = multicast, content = GRANT + OLT Signature is time capabilities + OLT public key+signature stamp encoded using OLT private kev REGISTER\_REQUEST(source addr. = ONU temp. MAC addr., content = PHY ID capabilities + ONU capabilities + Items written in italic echo of OLT capabilities + ONU permanent MAC address are encrypted using + ONU random temporary key) OLT public kev Contention REGISTER(dest\_addr = ONU temp. MAC addr, content = zone Items written in italic PHY ID list + echo of ONU capabilities + echo ONU are encrypted using permanent MAC address + 128 bit key) ONU random temporary key GATE(dest\_addr = ONU temp. MAC addr, content = GRANT) Items written in italic are encrypted using REGISTER\_ACK(content=echo of registered PHY ID) 128 bit key (stream? cipher, counter mode) Channel 802.1x is run optionally for authentication and established maybe to get a new key and encryption version. Encryption version is useful because one can then define a new encryption protocol and state it as a new encr. version if the initial is broken. After authentication keys and physical IDs may be

changed at any time by using control packets.