



Innovative R&D by NTT

# 802.1ad HW acceleration and MTU handling

Toshiaki Makita  
NTT Open Source Software Center

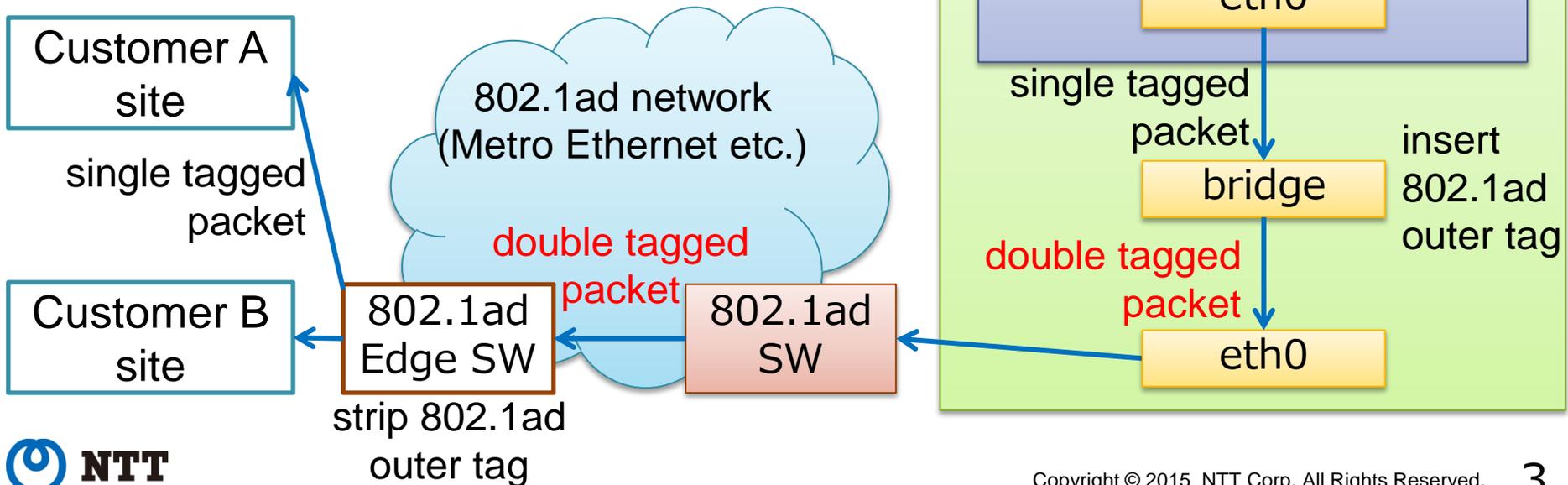
# Discussion topics



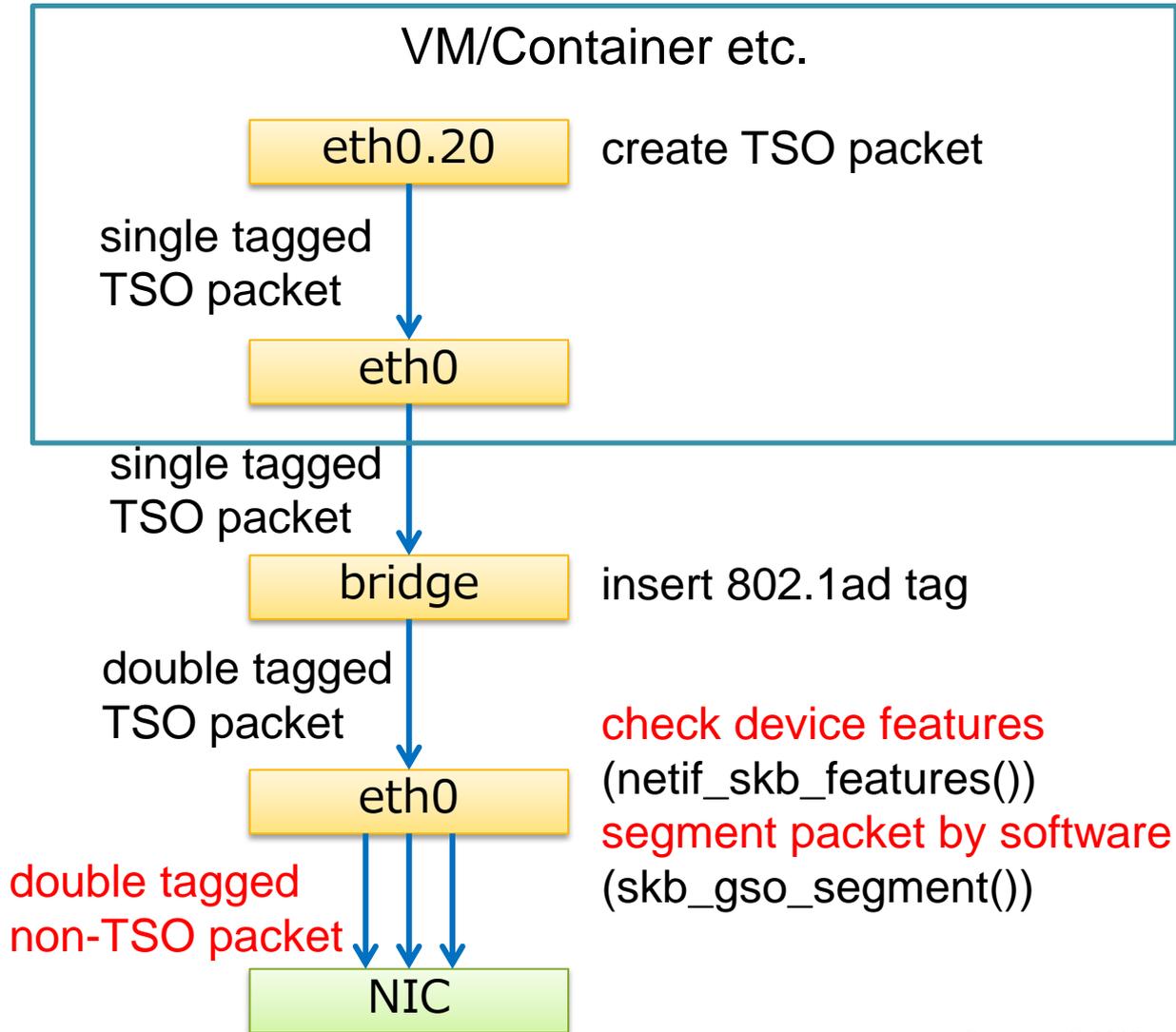
- **Discussion on 802.1ad issues and implementation to address them**
- **Offloading: TSO**
  - dev->vlan\_features cannot handle multiple vlans
- **Offloading: Vlan insert/strip**
  - There is no space for HW accelerating inner vlan
- **MTU (Tx/Rx buffer size)**
  - Received double tagged large vlan packets are dropped by default (oversize error)

# 802.1ad and Linux

- 802.1ad integrates Linux into existing Ethernet VPN (a.k.a. Metro Ethernet)
- 802.1ad tag (outer tag) is used to separate/identify Customers in Ethernet VPN



# TSO for stacked vlan



# NIC capability check for vlan

## Current checking for device features

```

netdev_features_t netif_skb_features(struct sk_buff *skb)
...
    features = netdev_intersect_features(features,
        apply_vlan_features
        for single tagged { dev->vlan_features |
                          NETIF_F_HW_VLAN_CTAG_TX |
                          NETIF_F_HW_VLAN_STAG_TX);

    if (protocol == htons(ETH_P_8021Q) || protocol == htons(ETH_P_8021AD))
        features = netdev_intersect_features(features,
            fixed_features_for_stacked_vlan
            (no TSO) { NETIF_F_SG |
                    NETIF_F_HIGHDMA |
                    NETIF_F_FRAGLIST |
                    NETIF_F_GEN_CSUM |
                    NETIF_F_HW_VLAN_CTAG_TX |
                    NETIF_F_HW_VLAN_STAG_TX);

finalize:
    if (dev->netdev_ops->ndo_features_check)
        features &= dev->netdev_ops->ndo_features_check(skb, dev,
                                                         features);
  
```

- **Need stacked vlan features for each physical device**
  - Some NICs seem to be able to handle double tagged TSO

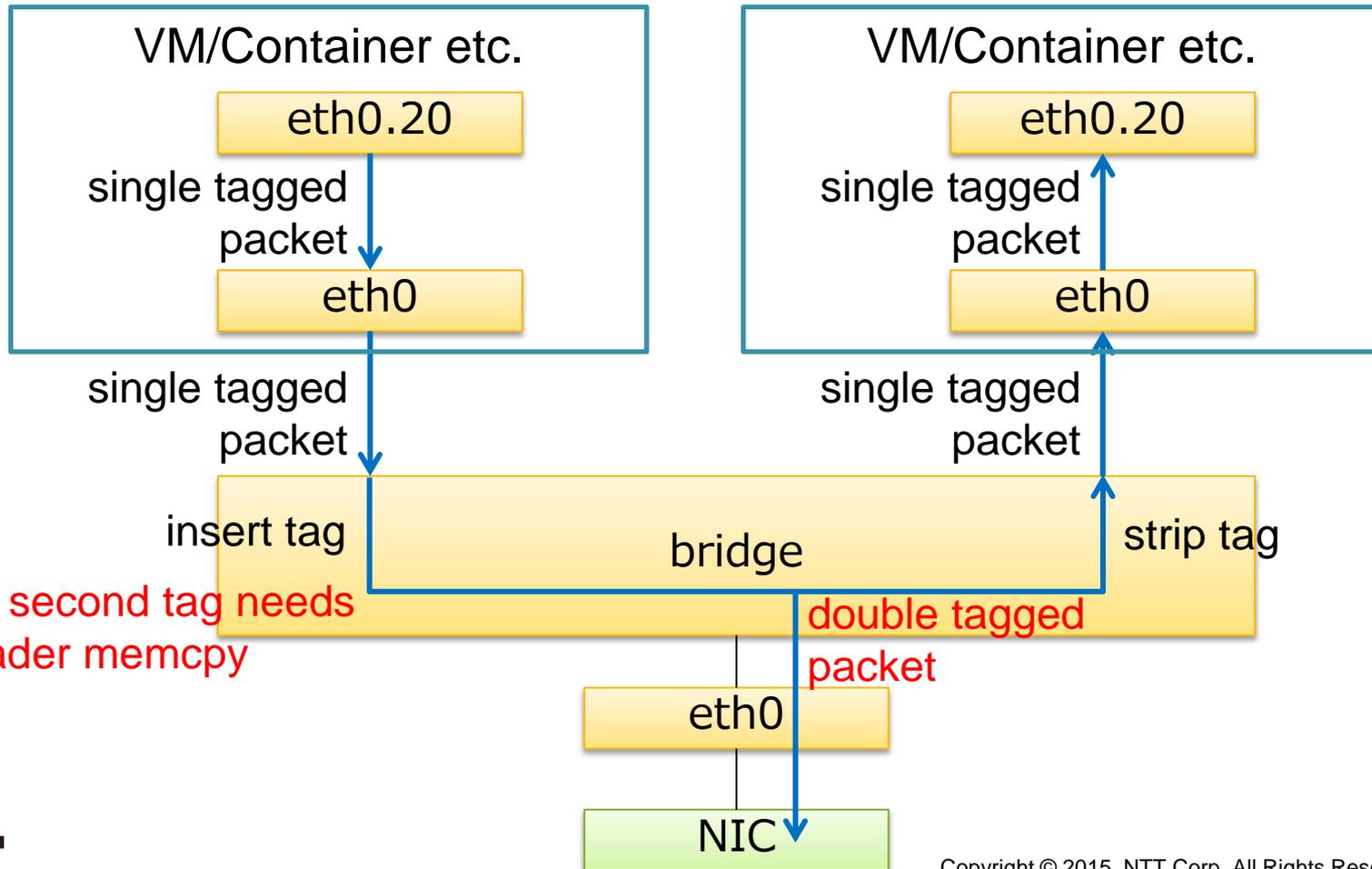
# Approach



- **Make dev->vlan\_features array? (vlan\_features[2])**
  - The number of tags are limited to array's size
- **Use ndo\_features\_check()?**
  - More flexible
  - Need dflt\_features\_check() to drop double tagged TSO by default?
- **Other thoughts?**

# insert/strip tag for stacked vlan

- Bridge internally insert/strip second tags when using `vlan_filtering`



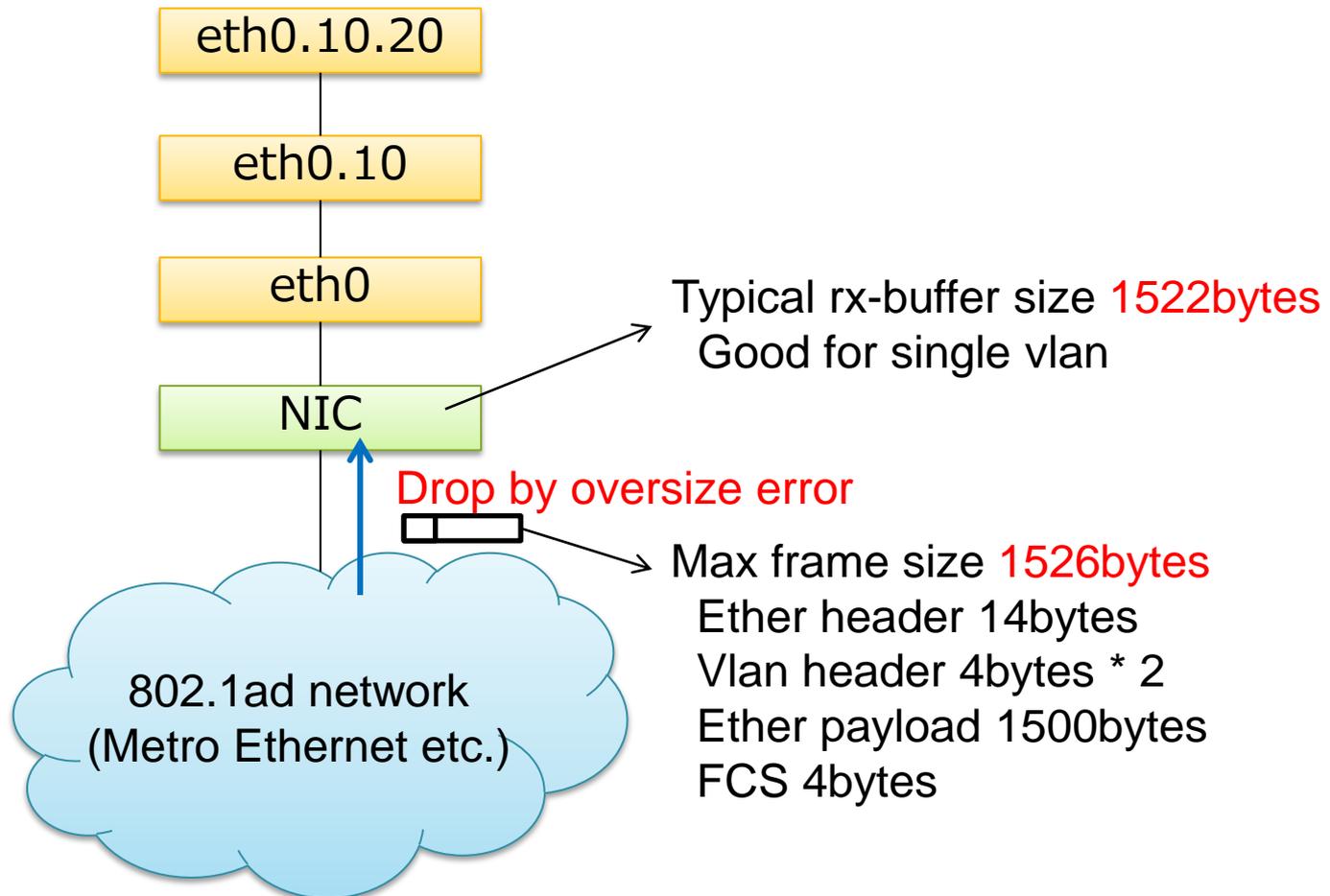
# Approach



- **Use `skb->cb[]`?**
  - Bridge-specific solution
  - Cannot be used for vlan offload on NIC
- **Make `skb->vlan_tci` array (`vlan_tci[2]`)?**
  - The number of tags are limited to array's size
- **Any other way?**

# MTU problem

- MTU-sized double tagged packet is dropped by default



# What's so problematic?



- **Looks like a strange random failure**
  - Ping is OK
  - TCP connection can be established
  - SSH is mostly OK
  - Only large packet is discarded
  - Hard to identify the root cause for users
- **Workaround varies driver by driver**
  - Setting MTU to 1504 would work in most cases
  - Sometimes 1508/9000 is needed depending on drivers
- **Bad experience for users...**

- **Always reserve 4bytes more room?**
  - Maybe overkill for a kind of corner case
  - Some driver changes behavior if buffer size is more than 1522
    - e.g. e1000e uses packet split descriptor
- **Automatically adjust buffer size on creating 802.1ad device**
  - introduce `ndo_enc_header_size()`
  - `bridge/vlan_dev` calls `lower_dev->ndo_enc_header_size()`
  - Can be used for appropriate tx buffer size
  - Other protocols (e.g. mpls) could also use it
- **Any other way?**