

## Application Note DT-AN-IP-2

# VLAN for DekTec Network Adapters

### 1. Introduction

VLAN (Virtual LAN) is a technology to segment a single physical network into multiple independent virtual networks. The VLANs are isolated from each other on the data link layer (layer 2): A networked application connected to one VLAN cannot communicate with an application on another VLAN<sup>1</sup>.

Reasons to use VLAN include the following:

- Logical portioning of network traffic;
- Security, to minimize the number of hosts to which broadcast packets are sent;
- Improve network performance.

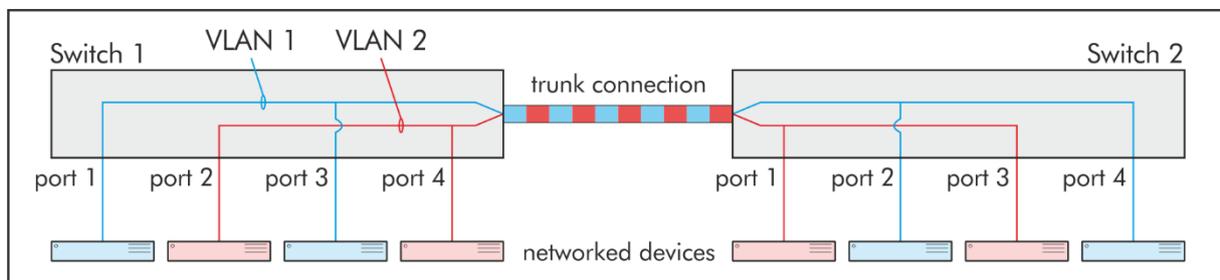
For example, in broadcast networking infrastructure, the video/audio/transport streams might be put on a certain VLAN (or a number of VLANs), while other network traffic such as management and control is put on a separate VLAN.

The protocol most commonly used today in configuring VLANs is IEEE 802.1Q3. This works by adding a so-called 802.1Q header (§3.3) to the Ethernet frame. Next to VLAN identifying information, this header contains a priority level to set the QoS required for the Ethernet frame. This can be used for traffic-flow management.

### 2. Using VLANs

#### 2.1. Using Switches to Create VLANs

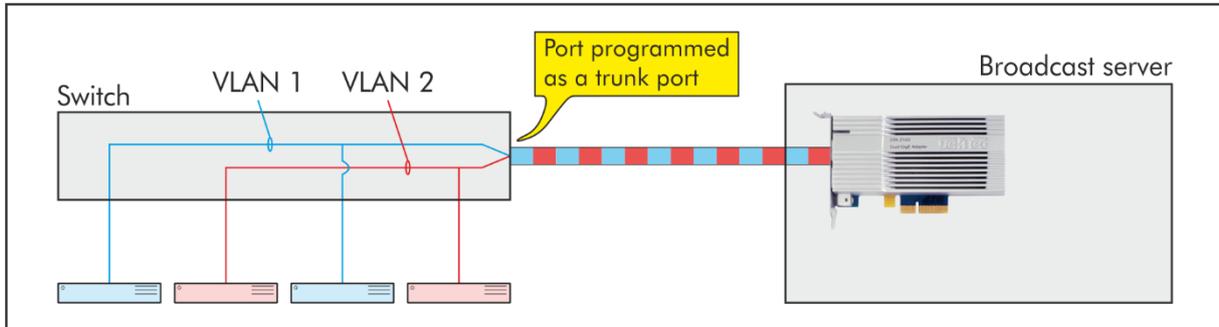
The usual way to segment a network into multiple VLANs is by using VLAN-aware switches. Each port of the switch is associated with a certain VLAN. For example, in the figure below, two VLANs are defined. The blue networked devices (PCs, servers, ...) are connected to VLAN 1, the red ones to VLAN 2. The devices itself are unaware of the VLAN they are connected too. However, the so-called "trunk-connection" between the two switches must be VLAN-aware, otherwise the switches cannot distinguish between the VLANs. For this reason, Ethernet packets on a trunk connection are tagged with a VLAN identifier.



<sup>1</sup> Hosts on different VLANs cannot communicate at the MAC layer, but they can communicate on the IP level through a Layer 3 router.

## 2.2. Supporting VLAN Directly from a NIC

Sometimes it is useful to support multiple VLANs directly from a Network Interface Card (NIC). For example, in a scenario with a server running several virtual machines with a single NIC, each VM may want to connect to a different VLAN. This can be achieved by programming the switch port to which the server NIC is connected as a “trunk port”.



DekTec network adapters support this way of working. Multiple VLANs can be configured, and in Windows each of them will show up as a separate connection in Control Panel’s Network Connections window. From the DekTec SDK it is straightforward to transmit or receive TS-over-IP streams to or from a specific VLAN.

The following DekTec network adapters support multiple VLANs:

Type Number	Description
DTA-160	Gigabit Ethernet and triple DVB-ASI ports for PCI bus
DTA-2160	Gigabit Ethernet and triple DVB-ASI ports for PCI Express bus
DTA-2162	Dual Gigabit Ethernet ports for PCI Express bus

## 3. The VLAN Protocol

### 3.1. VLAN Standards

The following standards related to VLAN are supported:

Standard	Description
IEEE 802.1Q	Defines a system of VLAN tagging for Ethernet frames and the accompanying procedures to be used by bridges and switches in handling tagged frames.
IEEE 802.1p	Quality of Service (QoS) prioritization scheme that uses a 3-bit field Priority Code Point (PCP) in the Ethernet frame header to differentiate network traffic.

### 3.2. Format of an Ethernet Frame

An Ethernet frame without VLAN tagging information contains the following fields:

Preamble	Destination address	Source address	EtherType	Payload	CRC
8 bytes	6 bytes	6 bytes	2 bytes	46..1500 bytes	4 bytes

For VLAN tagging, a 4-byte IEEE 802.1Q header is added between source address and EtherType:

Preamble	Destination address	Source address	802.1Q	EtherType	Payload	CRC
8 bytes	6 bytes	6 bytes	4 bytes	2 bytes	42..1500 bytes	4 bytes

### 3.3. Format of an IEEE 802.1Q Header

The IEEE 802.1Q header contains the following sub-fields:

Byte	Bits	Field	Field Name	Description
1,2	15..0	TPID	Tag Protocol Identifier	16-bit field set to 0x8100 in order to identify the frame as an IEEE 802.1Q-tagged frame.
3,4	15..13	PCP	Priority Code Point	3-bit field defining the frame priority level. IEEE 802.1p defines the priority levels.
	12	DEI	Drop Eligible Indicator	Not supported by the DekTec VLAN driver.
	11..0	VID	VLAN Identifier	12-bit field specifying the VLAN to which the frame belongs.

The VLAN Identifier<sup>2</sup> uniquely identifies a VLAN. The maximum number of VLANs is 4094. VLAN IDs 0 and 0xFFF are reserved. VLAN ID 0 is used for untagged packets, i.e. Ethernet packets without VLAN information. The default VLAN usually has a VLAN ID of 1.

### 3.4. IEEE 802.1p Priority Levels

The PCP field in the IEEE 802.1Q header (§3.3) encodes the priority of Ethernet frames. It can be used by Ethernet switches to implement QoS management. The priority levels are defined in IEEE 802.1p as follows:

PCP	Priority	Traffic Type	Comment
1	0 (lowest)	BK = Background	
0	1	BE = Best Effort	
2	2	EE = Excellent Effort	
3	3	CA = Critical Applications	
4	4	VI = Video	Jitter should remain less than 100ms
5	5	VO = Voice	Jitter should remain less than 10ms
6	6	IC = Internetwork Control	
7	7 (highest)	NC = Network Control	

QoS management with priority levels is independent from the virtual LAN function. However, the two are coupled because they share the same sub-header. It is perfectly fine to use QoS management with the PCP field, but not the VLAN function. In that case, VLAN ID must be set to 0.

---

<sup>2</sup> The standard abbreviates VLAN Identifier by VID, but we will use the more descriptive VLAN ID.

## 4. Installing VLAN for DekTec Adapters

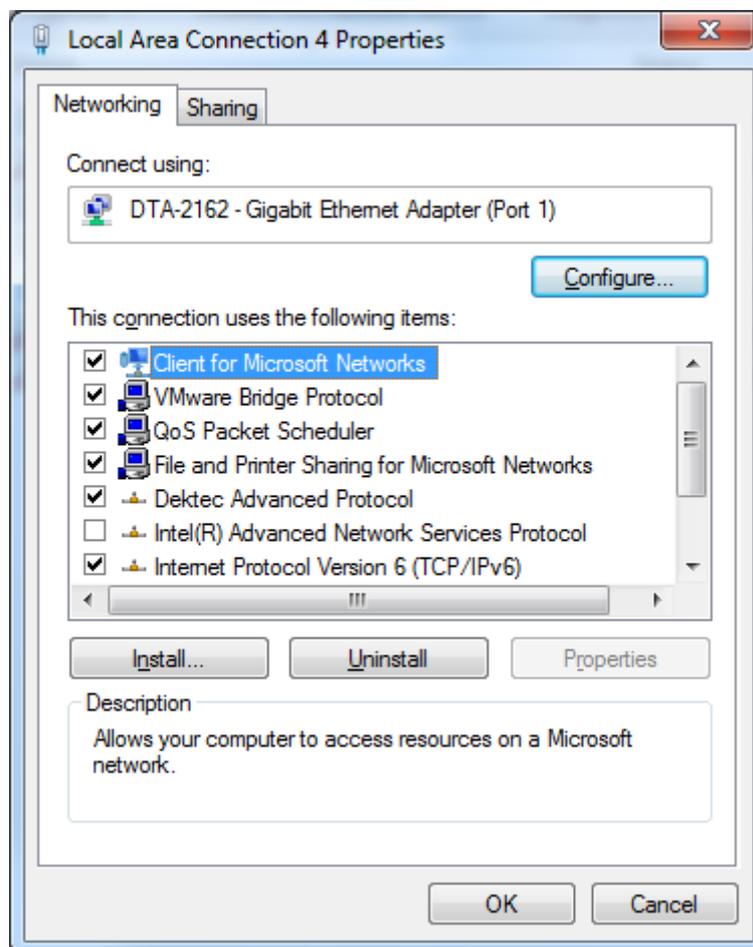
### 4.1. Windows

To use VLAN on a Windows host, you have to install the DekTec Advanced Protocol driver [DtaNwAp64](#). This driver is included in [Dta.zip](#), available from the DekTec website, section *Downloads*, page *Drivers and SDKs*. Running [DtaInstall.exe](#) from this zip file, which you probably have done already to install the DekTec Windows drivers, stores the advanced protocol driver files in:

[Program Files \(x86\)\DekTec\Drivers\DtaNwAp64](#).

To actually install [DtaNwAp64](#), perform the following steps:

- 1) In *Control Panel*, select *Network and Sharing Center* (or similar, depending on your version of Windows) to show the *Network Connections* window.
- 2) Identify the Local Area Connection, right click and select *Properties*.



- 3) Click the *Install* button.
- 4) Select *Protocol* and push the *Add* button.
- 5) Push the *Have Disk* button and locate the directory where the driver is located: [Program Files \(x86\)\DekTec\Drivers\DtaNwAp64](#) (or similar).
- 6) Select [DtaNwAp64.inf](#), press *OK*, and the driver will be installed.

After the driver is installed, you can add VLANs by selecting the *Configure* button in the *Properties* window (see screenshot at step 2 above). In the *VLAN* tab, you can add a new VLAN with a specified VLAN ID and description. Please note that adding a VLAN may take some time. The dialog should not be closed before installation is completed.

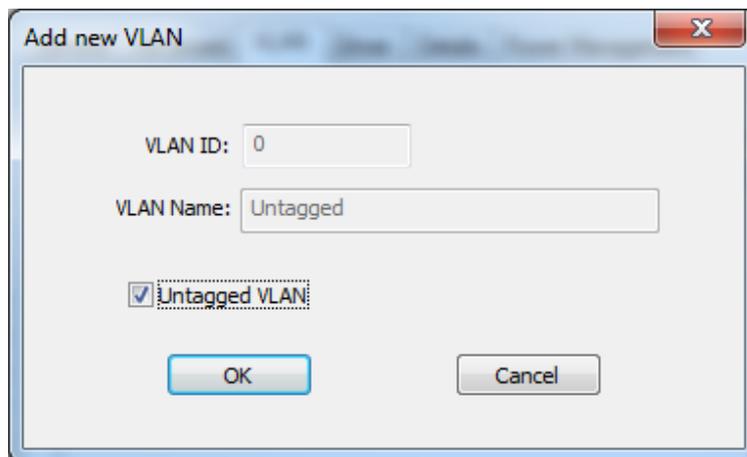
A VLAN can be removed by selecting it and pushing the *Remove* button.

If you don't see the VLAN tab, check whether the DekTec Advanced Protocol driver is listed in the protocol items of the connection (see the screenshot on the previous page). If it is not listed, you may try to reinstall the protocol driver. If the protocol driver is installed, and you still don't see the VLAN tab, check whether you have the latest version of **DtaNw** (DekTec network driver) installed.

A new (virtual) network adapter is created for each VLAN that is added. This network adapter can only be removed with the *Remove* button in the VLAN tab. The figure below shows an example: Local Area Connection 4 is the physical network connection representing port 1 of a DTA-2162. Two virtual network adapters have been created, one for VLAN 5 and one for VLAN 10.

Name	Status	Device Name
Local Area Connection 4	Enabled	DTA-2162 - Gigabit Ethernet Adapter (Port 1)
Local Area Connection 5	Identifying...	VLAN5:DTA-2162 - Gigabit Ethernet Adapter (Port 1)
Local Area Connection 6	Identifying...	VLAN10:DTA-2162 - Gigabit Ethernet Adapter (Port 1)

If you add a VLAN, the protocol bindings of the main network connection are disabled and you can't use this connection to send/receive network packets. If you want to receive the network packets that are not directed to a specific VLAN, you have to add an "Untagged VLAN" by selecting the 'Untagged VLAN' check box in the *Add new VLAN* dialog:



## 4.2. Linux

Recent Linux versions support VLAN out of the box. If you have an older Linux version, you must recompile the Linux kernel with the VLAN option enabled.

To use VLAN on Linux, you need to add VLANs and specify a VLAN ID for each of them. You can do this manually, or automatically during system start.

### 4.2.1. Manually Creating VLANs

Use the **vconfig** command from the console, e.g.

```
vconfig add eth0 5
```

This **vconfig** command creates a VLAN device on **eth0** with interface name **eth0.5**. Using this interface name, you can now use all the network tools in the same way as you use them with a regular network adapter without VLAN.

To activate the VLAN interface with VLAN ID 5, use: **ifconfig eth0.5 up**

#### 4.2.2. Automatically Creating VLANs at System Start

You can set up your system so that the VLAN devices for a network device are automatically created during system start. How to configure this is distribution dependent. Below a short description is provided for some popular Linux distributions. For other distributions, please refer to the documentation for VLAN configuration details.

For Debian, Ubuntu and derivative distributions you have to add the virtual VLAN device to the file [/etc/network/interfaces](#). For example, add the following three lines to automatically create a VLAN with VLAN ID 5 on network device **eth0** at boot time.

```
auto vlan5
iface vlan5 inet dhcp
vlan_raw_device eth0
```

Obviously, **eth0** has to be replaced by the network device on your system, and **vlan5** by the required VLAN interface name. Note that the IP address for **vlan5** is obtained via DHCP.

For RHEL and CentOS you have to create a new file in [/etc/sysconfig/network-scripts](#). The filename has to be the physical interface name, followed by a '.', followed by the VLAN ID number. For example, for VLAN ID 5, physical interface **eth0** and dynamic IP configuration with DHCP, the filename should be **ifcfg-eth0.5**, and the contents should be as follows:

```
DEVICE=eth0.5
BOOTPROTO=dhcp
ONBOOT=yes
VLAN=yes
```

## 5. Using VLAN from DTAPI

To use VLAN with DTAPI is relatively straightforward. You just have to let DTAPI know which VLAN ID you want to use to transmit (or receive) the Ethernet packets to (or from). This can be done using the **SetIpPars** function with a **DtTsIpPars** argument.

The **DtTsIpPars** structure contains a member **m\_VlanId** that must be set to the required VLAN ID. If you want to send 'untagged' packets, this member must be set to 0. The VLAN ID set in **m\_VlanId** must be one of the VLAN IDs created during the VLAN setup.

You can also set the IEEE 802.1p priority field in the **m\_Priority** field. DTAPI will create Ethernet frames with the priority value stuffed in the PCP field of the IEEE 802.1Q header.

Please find below a code snippet that shows setting up a **DtTsIpPars** structure for transmitting a stream on VLAN ID 12 with IEEE 802.1p priority level 4 (Video with <100ms latency and jitter).

```
DtTsIpPars TsIpPars;           // TS-over-IP parameters used for Tx

// Set up destination IP address and port
TsIpPars.m_Ip[0] = 192;
TsIpPars.m_Ip[1] = 168;
TsIpPars.m_Ip[2] = 1;
TsIpPars.m_Ip[3] = 100;
TsIpPars.m_Port = 12345;

// Set up VLAN parameters
TsIpPars.m_VlanId = 12;       // Sets VLAN ID to 12
TsIpPars.m_VlanPriority = 4;  // Sets IEEE 802.1p priority to 4

// Set up other parameters
TsIpPars.m_TimeToLive = 128;
TsIpPars.m_NumTpPerIp = 7;
TsIpPars.m_Protocol = DTAPI_PROTO_UDP;
TsIpPars.m_FecMode = DTAPI_FEC_DISABLE;
TsIpPars.m_Flags = DTAPI_IP_V4;
TsIpPars.m_Mode = DTAPI_IP_NORMAL;

// Set the TS-over-IP parameters in the output channel.
// This assumes TsOutp is a DtOutpChannel object that has been
// attached to the hardware (e.g. DTA-2162)
TsOutp.SetIpPars(TsIpPars);
```