TiVo Connect Automatic Machine Discovery Protocol Specification

1 Introduction

This document describes how "machines" (TiVo DVRs and/or PCs) running TiVo Connect software automatically "discover" each other's presence on the network.

1.1 Audience

Developers of TiVo Connect software applications.

1.2 Disclaimers

The TiVo Connect Automatic Machine Discovery protocol specification is provided solely for informational purposes. It is intended solely to enable you to write programs that allow you use your DVR to listen to music files or view photograph files that reside on your home computer for personal, noncommercial use. The TiVo Connect Automatic Machine Discovery protocol is not intended to be used for listening to or viewing music or photographs from third party sources including, but not limited to, streaming, webcasting, or any other form of transmission of content. Listening to music or viewing photographs from third party sources could subject your DVR and/or home computer to harm and may infringe the copyrights of third parties.

TIVO IS NOT RESPONSIBLE FOR ANY HARM TO YOUR DVR, COMPUTER, OR OTHER DEVICES RESULTING FROM YOUR USE OF THE TIVO CONNECT AUTOMATIC MACHINE DISCOVERY PROTOCOL. TIVO IS NOT RESPONSIBLE FOR, NOR HAS ANY CONTROL OVER, THE CONTENT OF ANY MUSIC OR PHOTOGRAPHS ACCESSED USING A TIVO DVR. UNAUTHORIZED COPYING OR DISTRIBUTION OF COPYRIGHTED WORKS IS AN INFRINGEMENT OF THE COPYRIGHT HOLDERS' RIGHTS. TIVO RESERVES THE RIGHT TO TERMINATE THE ACCOUNTS OF USERS OF ANY TIVO SOFTWARE WHO INFRINGE THE COPYRIGHTS OF OTHERS.

1.3 References

RFC 768 – User Datagram Protocol
RFC 793 – Transmission Control Protocol
RFC 922 – Broadcasting Internet Datagrams in the Presence of Subnets

2 Overview

The TiVo Connect Automatic Machine Discovery Protocol allows two or more machines running TiVo Connect software (or simply TCM for "TiVo Connect Machine") to discover each other by periodically exchanging UDP and/or TCP packets of TiVo-defined identifying information. Because of their periodic nature, such packets are typically referred to as "beacons".

Within this protocol, beacons transmitted via UDP are always "broadcast-based", meaning that the packets are sent blindly onto the local network, with absolutely no special acknowledgment or other handshaking required. Interested TCMs need simply "listen" passively in order to detect other TCMs. (Imagine a lighthouse lamp rotating over a foggy bay, announcing its presence to any ships close enough to see it.)

On the other hand, beacons transmitted via TCP are "connection-based", requiring explicit two-way handshaking. This approach is required to overcome limitations of certain network configurations in which the one-way broadcasting of UDP packets isn't effective – due to network topology or policy, preventing certain TCMs from being able to "hear" each other. (Imagine shining an infinitely powerful flashlight directly at someone lost in a dark cavern.)

In addition to machine identification, the protocol also has a simple mechanism for announcing the availability of services provided by each TCM. From the perspective of a single TCM, once identity and service availability have been established, the discovery phase is largely complete. From then on, other protocols (such as HTTP) can kick in, normally as part of accessing and/or providing various services. However, see section 3.5 for details about how updates to the discovered information occur.

---

3 Details

There are three main components to the workings of the protocol.

- Beacon Packet Data Format
- Broadcast-based Discovery
- Connection-based Discovery

The packet data format is common to all other aspects of the protocol, while the broadcast-based and connection-based discovery mechanisms, although similar, each have their own associated details. Every TCM must be prepared to participate in broadcast- and connection-based discovery simultaneously.

Regardless of the method used to transmit beacon packets, each participating TCM maintains an internal list of all other TCMs from which it has heard. Records in this list are updated whenever new information arrives. Records for TCMs that have not been heard from recently (or whose departures have been explicitly detected) are eventually cycled off this list.

In this way, whenever further communication is needed, the set of networked machines able to "TiVo Connect", as well as their available services, can be known at any given moment with no need to query the network.

3.1 Beacon Packet Data Format

The data contained in each beacon packet is made up of an intentionally small amount of information (assumed to always fit within a single UDP or TCP packet), formatted in ASCII as simple name/value pairs, one per line:

```
tivoconnect=<number>
```

```
method=<method>
platform=<type>[/<sub-type>]
machine=<string>
identity=<string>
services=<name>[:<port>][/<protocol>], ...
swversion=<string>
```

The <name> portion of each pair is case-insensitive, so "PLATFORM" and "pLatFOrm" both identify the same value.

In the future, this format can be extended without disturbing any existing software through the addition of pairs with new names. As such, care should by taken in current software to avoid interpreting this data with any assumptions about the particular number, collection, or ordering of lines within the packet (with one exception for "tivoconnect", described in section 3.1.1).

TCMs encountering any unrecognized value (including an empty line) should always simply ignore it, moving on the to the next line (if any).

3.1.1 tivoconnect

This value indicates the particular flavor of TiVo Connect Automatic Machine Discovery Protocol supported by the originating TCM. For now, <number> should always be "1".

As the one exception to the previously stated ordering assumptions, this value must always appear at the very beginning of every packet so the sequence "tivoconnect" (in the first 11 character positions) can serve as an identifying "signature". Note that "TIVOCONNECT" and "tiVoCOnNecT" are both examples of valid signatures.

This value is not optional.

3.1.2 method

This value indicates the way in which the packet was transmitted, where <method> should be one of the following values:

- broadcast (for packets sent using UDP)
- connected (for packets sent using TCP)

This value is not optional.

3.1.3 platform

This value indicates what kind of TCM sent the packet, where <type> should be one of the following values:

- tcd (for TiVo DVR beacons)
- pc (for Windows computer beacons)

The use of unique values for <type> creates platform "namespaces", within which various values for the optional <sub-type> portion can be freely determined by the associated software development groups without risk of conflict. Future development efforts should use their own value for <type> (e.g., "c64" for Commodore-64 software applications).

This value is not optional.

3.1.4 machine

This value contains human readable text, naming the TCM, suitable for display to the user.

Windows computer beacons should set <string> to the Windows computer name. TiVo DVR beacons should set <string> to the name of the DVR (e.g., "The Upstairs Master Bedroom Closet TiVo"). Packets originating from other platforms should contain a similar suitable string.

This value is optional (or can be left blank) if the machine does not have a name. However, a name is highly recommended since it can be used by software to enhance the user's experience.

3.1.5 identity

This value should be unique to the originating TCM (perhaps even globally unique, but certainly unique across the local network). This information is intended to allow TCMs to unambiguously identify each other even when their names or IP addresses have changed.

TiVo DVR beacons should set <string> to the DVR's serial number. Windows computer packets should set <string> to a GUID (generated once and stored in the registry), formatted using the StringFromGUID2() function of the Windows API.

This value is not optional.

3.1.6 services

This value provides a comma-delimited list of entries indicating the availability of services and optionally the port numbers and protocols through which they communicate.

This value is optional (or can be left blank) as not every TCM will necessarily provide any services.

3.1.7 swversion

This value describes the "primary" software running on the TCM. There is no required format for <string>.

This value is optional.

3.2 Example Packets

The ASCII data in a typical beacon packet might look like this.

```
tivoconnect=1
method=broadcast
platform=pc/win-nt
machine=FREDS-PC
identity={D936E980-79E3-11D6-A84A-00045A43EEE7}
services=FooService:1234,BarService:4321
```

Note that the following packet is equivalent, even though the case of some names and the ordering of lines is different.

```
TivoConnect=1
IDENTity={D936E980-79E3-11D6-A84A-00045A43EEE7}
maCHine=FREDS-PC
sERVices=FooService:1234,BarService:4321
PLATFORM=pc/win-nt
Method=broadcast
```

3.3 Broadcast-based Discovery

Normally, UDP beacons are sent periodically to the broadcast address of the local network, on port 2190 (registered to TiVo).

Upon startup (meaning TiVo DVR boot up or activation of TiVo Connect software running on some other hardware) a TCM broadcasts, for a short period of time (say, 30 seconds) a number of redundant packets in "high frequency mode" (perhaps every 5 seconds). After this initial period, the TCM drops into "low frequency mode" broadcasting at a reduced rate (maybe only once every minute). The initial high-frequency mode allows listeners several chances to quickly detect TCMs that have just arrived, while the eventual low-frequency broadcasts should be infrequent enough to not overly burden the network – even in the presence of many TCMs.

With broadcast-based discovery, every TCM must be prepared to accept redundant packets, and should always assume they contain the latest information about the originating TCM. Again, keep in mind that "connection-less" UDP beacons are sent blindly onto the network – with no guarantee that they'll ever be received. The periodic and redundant nature of the broadcast-based discovery mechanism is necessary for it to work effectively.

3.4 Connection-based Discovery

Normally, TCP beacons are exchanged between two specific TCMs, one TCM initiating the connection on port 2190.

With the IP address (typically supplied by a user) associated with some "target" TCM, the initiating TCM establishes a TCP connection with the target TCM and then transmits a single packet. The target TCM responds with its own packet. From this point on, the connection is maintained (normally "quiet", with no additional packets sent) until either TCM drops the connection. Although TCP beacons need only be exchanged once upon initial connection, each TCM should be prepared to accept and process redundant packets from the other, just as with broadcast-based discovery.

3.5 Update Procedure

When any information about a TCM changes, the affected TCM should switch back to high-frequency mode (again for 30 seconds, just as upon startup) broadcasting several packets containing the new information. This will allow the changes to propagate quickly to any other TCMs participating in broadcast-based discovery. In order to also propagate the changes to those TCMs participating in connection-based discovery, a single packet containing the new information should also be sent to each currently connected TCM (regardless of whether the affected TCM initiated the connection or not).

---

4 Caveats

4.1 Machine Arrival

When a new TCM arrives on the local network, its initial high-frequency broadcasts will allow other TCMs participating in broadcast-based discovery to detect it almost immediately. However, the new TCM itself may not immediately detect other TCMs that have already dropped into low-frequency mode. To combat this, it is suggested any TCM detecting a new TCM temporarily switch back to high-frequency mode itself (the recommended duration is again 30 seconds). In this way, the new TCM will to able to quickly discover the other TCMs as well.

4.2 Service Availability

Whenever a service becomes available on a TCM, a corresponding entry should be added to the "services" value in any subsequently transmitted beacon packets. Likewise, when the service becomes unavailable, the entry should be removed. Whenever either occurs, the affected TCM should promptly execute the update procedure described in section 3.5.