DCCP WG
Internet-Draft
Intended status: Proposed Standard
Expires: July 18, 2008
Updates: RFC 4340

G.Fairhurst
University of Aberdeen

September 29, 2008

The DCCP Service Code
draft-ietf-dccp-serv-codes-08.txt

Status of this Memo

Abstract

This document describes the usage of Service Codes by the Datagram Congestion Control Protocol, RFC 4340. It motivates the setting of a Service Code by applications. Service Codes provide a method to identify the intended service/application to process a DCCP connection request. This provides improved flexibility in the use and assignment of port numbers for connection multiplexing. The use of a DCCP Service Code can also enable more explicit coordination of services with middleboxes (e.g. network address translators and firewalls). This document updates the specification provided in RFC 4340.

Table of Contents

1. Introduction

DCCP specifies a Service Code as a 4-byte value (32 bits) that describes the application-level service to which a client application wishes to connect ([RFC4340], section 8.1.2). A Service Code identifies the protocol (or a standard profile, e.g. [ID.RTP]) to be used at the application layer. It is not intended to be used to specify a variant of an application, or a specific variant of a protocol (Section 2.2).

The Service Code mechanism allows an application to declare the set of services that are associated with server port numbers. This can affect how an application interacts with DCCP. It allows decoupling the role of port numbers to indicate a desired service from the role in connection demultiplexing and state management. A DCCP application identifies the requested service by the Service Code value in a DCCP-Request packet. Each application therefore associates one or more Service Codes with each listening port ([RFC4340], section 8.1.2).

The use of Service Codes can assist in identifying the intended service by a firewall and may assist other middleboxes (e.g., a proxy server, network address translator (NAT) [RFC2663]). Middleboxes that desire to identify the type of data a flow claims to transport, should utilize the Service Code for this purpose. When consistently used, the Service Code can provide a more specific indication of the actual service (e.g. indicating the type of multimedia flow, or intended application behaviour).

The more flexible use of server ports can also offer benefit to applications where servers need to handle very large numbers of simultaneous open ports to the same service.

RFC 4340 omits to describe the motivation behind Service Codes, nor does it properly describe how Well Known and Registered server ports relate to Service Codes. The intent of this document is to clarify these issues.

1.1. History

It is simplest to understand the motivation for defining Service Codes by describing the history of the DCCP protocol.

Most current Internet transport protocols (TCP [RFC793], UDP [RFC768], SCTP [RFC4960], UDP-Lite [RFC3828]) used "Published" port numbers from the Well Known or registered number spaces [RFC814]. These 16-bit values indicate the application service associated with a connection or message. The server port must be known to the client

to allow a connection to be established. This may be achieved using out-of-band signaling (e.g. described using SDP [RFC4566]), but more commonly a Published port is allocated to a particular protocol or application; for example HTTP commonly uses port 80 and SMTP commonly uses port 25. Making a port number Published [RFC1122] involves registration with the Internet Assigned Numbers Authority (IANA), which includes defining a service by a unique keyword and reserving a port number from among a fixed pool [IANA].

In the earliest draft of DCCP, the authors wanted to address the issue of Published ports in a future-proof manner, since this method suffers from several problems:

- The port space is not sufficiently large for ports to be easily allocated (e.g. in an unregulated manner). Thus, many applications operate using unregistered ports, possibly colliding with use by other applications.

- The use of port-based firewalls encourages application-writers to disguise one application as another in an attempt to bypass firewall filter rules. This motivates firewall writers to use deep packet inspection in an attempt to identify the service associated with a port number.

- ISPs often deploy transparent proxies, primarily to improve performance and reduce costs. For example, TCP requests destined to TCP port 80 are often redirected to a web proxy.

These issues are coupled. When applications collide on the same Published, but unregistered port, there is no simple way for network security equipment to tell them apart, with the likelihood of introducing problems with interaction of features.

There is little that a transport protocol designer can do about applications that attempt to masquerade as other applications. For ones that are not attempting to hide, the problem may be simply that they cannot trivially obtain a Published port. Ideally, it should be sufficiently easy that every application-writer can request a Well Known or registered port and receive one instantly with no questions asked. The 16-bit port space traditionally used is not large enough to support such a trivial allocation of ports.

Thus, the design of DCCP sought an alternative solution. The idea was simple. A 32-bit server port space should be sufficiently large that it enables use of very simple allocation policies. However, overhead considerations made a 32-bit port value undesirable (DCCP needed to be useful for low rate applications).

The solution in DCCP to this problem was to use a 32-bit Service Code [RFC4340] that is included only in the DCCP-Request packet. The use of a 32-bit value was intended to make it trivially simply to obtain a unique value for each application. Placing the value in a DCCP-Request packet, requires no additional overhead for the actual data flow. It is however sufficient for both the end systems, and provides any stateful middleboxes along the path with additional information to understand what applications are being used.

Early discussion of the DCCP protocol considered an alternative to the use of traditional ports; instead it was suggested that a client used a 32-bit identifier to uniquely identify each connection. The server listened on a socket bound only to a Service Code. This solution was unambiguous; the Service Code was the only identifier for a listening socket at the server side. The DCCP client included a Service Code in the request, allowing it to reach the corresponding listening application. One downside was that this prevented deployment of two servers for the same service on a single machine, something that is trivial with ports. The design also suffered from the downside of being sufficiently different from existing protocols that there were concerns that it would hinder the use of DCCP through NATs and other middleboxes.

RFC 4340 abandoned the use of a 32-bit connection identifier in favor of two traditional 16-bit port values, one chosen by the server and one by the client. This allows middleboxes to utilize similar techniques for DCCP, UDP, TCP, etc. However, it introduced a new problem: "How does the server port relate to the Service Code?" The

intent was that the Service Code identified the application or protocol using DCCP, providing middleboxes with information about the intended use of a connection, and that the pair of ports effectively formed a 32-bit connection identifier, which was unique between a pair of end-systems.

The large number of available unique Service Code values allows all applications to be assigned a unique Service Code. However, there remains a current problem: The server port is chosen by the server, but the client needs to know this to establish a connection. It was undesirable to mandate out-of-band communication to discover the server port. A solution is to register DCCP server ports. The limited availability of DCCP server ports appears to contradict the benefits of DCCP Service Codes, because although it may be trivial to obtain a Service Code, it has not traditionally been trivial to obtain a registered port from IANA and in the long-run it may not be possible to uniquely allocate a unique registered DCCP port to new applications. As port numbers become scarce, this motivates the need to associate more than one Service Code with a listening port (e.g.

two different applications could be assigned the same server port, and need to run on the same host at the same time, differentiated by their different associated Service Codes.

Service Codes provide flexibility in the way clients identify the server application to which they wish to communicate. The mechanism allows a server to associate a set of server ports with a service. The set may be common with other services available at the same server host, allowing a larger number of concurrent connections for a particular service than possible when the service is identified by a single Published port number.

There has been confusion concerning how server ports relate to Service Codes. The goal of this document is to clarify this and the issues concerning the use of Service Codes.

RFC4340 states that Service Codes are not intended to be DCCP-specific. Service Codes, or similar concepts may therefore also be useful to other IETF transport protocols.

1.2. Conventions used in this document

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT",

"SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

2. An Architecture for Service Codes

DCCP defines the use of a combination of ports and Service Codes to identify the server application ([RFC4340], section 8.1.2). These are described in the following Sections.

2.1. IANA Port Numbers

In DCCP, the packets belonging to a connection are de-multiplexed based on a combination of four values {source IP address, source port, dest IP address, dest port}, as in TCP. An endpoint address is associated with a port number, (e.g. forming a socket); and a pair of associations uniquely identifies each connection. Ports provide the fundamental per-packet de-multiplexing function.

The Internet Assigned Numbers Authority currently manages the set of globally reserved port numbers [IANA]. The source port associated with a connection request, often known as the "ephemeral port", is traditionally in the range 49152-65535, and also includes the range 1024-49151. The value used for the ephemeral port is usually chosen by the client operating system. It has been suggested that a

randomized choice of port number value can help defend against "blind" attacks [ID.Rand] in TCP. This method may be applicable to other IETF-defined transport protocols, including DCCP.

Traditionally, the destination (server) port value associated with a service is determined either by an operating system index to a copy of the IANA table (e.g., getportbyname() in Unix, which indexes the /etc/services file), or directly mapped by the application.

The UDP and TCP port number space: 0..65535, is split into three ranges [RFC2780]:

- 0..1023 "Well Known", also called "system" ports,
- 1024..49151 "registered", also called "user" ports,
- 49152..65535 "dynamic", also called "private" ports.

DCCP supports Well Known and registered ports. These are allocated in the DCCP IANA port numbers registry ([RFC4340], Section 19.9). Each registered DCCP port MUST be associated with at least one pre-defined Service Code.

Applications that do not need to use a server port in the Well Known or registered range SHOULD use a dynamic server port (i.e. that does not require to be registered in the DCCP port registry). Clients can identify the server port value for the services to which they wish to connect using a range of methods. One common method is by reception of a SDP record (Section 2.6) exchanged out-of-band (e.g. using SIP [RFC3261] or RTSP [RFC2326]). DNS SRV resource records also provide a way to identify a server port for a particular service based on the services string name [RFC2782].

Applications that do not use out-of-band signalling can still communicate, providing that both client and server agree the port value to be used. This eliminates the need for each registered Service Code to be allocated an IANA-assigned server port (see also Section 2.7).

2.2. DCCP Service Code Values

DCCP specifies a 4 byte Service Code ([RFC4340], section 8.1.2) represented in one of three forms: a decimal number (the canonical method), a four character ASCII string, or an eight digit hexadecimal number.

The Service Code identifies the application-level service to which a client application wishes to connect. Examples of services are RTP [ID.RTP], TIME (this document), ECHO (this document). In a different example, DTLS [RFC5238] provides a transport-service (not an application-layer service), therefore applications using DTLS are individually identified by a set of corresponding Service Code values.

Endpoints MUST associate a Service Code with every DCCP socket [RFC4340], both actively and passively opened. The application will generally supply this Service Code. A single passive listening port may be associated with more than one Service Code value. The set of Service Codes could be associated with one or more server

applications. This permits a more flexible correspondence between services and port numbers than possible using the corresponding socket pair (4-tuple of layer-3 addresses and layer-4 ports). In the currently defined set of packet types, the Service Code value is present only in DCCP-Request ([RFC4340], section 5.2) and DCCP-Response packets ([RFC4340], section 5.3). Note new DCCP packet types (e.g. [ID.Simul]) could also carry a Service Code value.

2.2.1. New versions of Applications or Protocols

Applications/protocols that provide version negotiation or indication in the protocol operating over DCCP do not require a new server port or new Service Code for each new protocol version. New versions of such applications/protocols SHOULD continue to use the same Service Code. If the application developers feel that the new version provides significant new capabilities (e.g. that will change the behavior of middleboxes), they MAY allocate a new Service Code associated with the same or a different set of Well Known ports. If the new Service Code is associated with a Well Known or registered port, the DCCP Ports registry MUST also be updated to include the new Service Code value, but MAY share the same server port assignment(s).

2.3. Service Code Registry

The set of registered Service Codes specified for use within the general Internet are defined in an IANA-controlled name space. IANA manages new allocations of Service Codes in this space ([RFC4340]). Private Service Codes are not centrally allocated and are denoted by the decimal range 1056964608-1073741823 (i.e. 32-bit values with the high-order byte equal to a value of 63, corresponding to the ASCII character '?').

Associations of Service Code with Well Known Ports are also defined in the IANA DCCP Port Registry (section 2.1).

2.4. Zero Service Code

A Service Code of zero is "permanently reserved (it represents the absence of a meaningful Service Code)" [RFC4340]. This indicates that no application information was provided. RFC 4340 states that applications MAY be associated with this Service Code in the same way as other Service Code values. This use is permitted for any server port.

This document clarifies section 19.8 of RFC 4340, by adding the following:

"Applications SHOULD NOT use a Service Code of zero.

Application writers that need a temporary Service Code value SHOULD choose a value from the private range (section 2.3).

Applications intended for deployment in the Internet are encouraged to use an IANA-defined Service Code. If no specific Service Code exists, they SHOULD request a new assignment from the IANA."

2.5. Invalid Service Code

RFC4340 defines the Service Code value of 0xFFFFFFFF as Invalid. This is provided so implementations can use a special four-byte value to indicate "no valid Service Code". Implementations MUST NOT accept a DCCP-Request with this value, and SHOULD NOT allow applications to bind to this Service Code value [RFC4340].

2.6. SDP for describing Service Codes

Methods that currently signal destination port numbers, such as the Session Description Protocol (SDP) [RFC4566] require extension to support DCCP Service Codes [ID.RTP].

2.7. A method to hash the Service Code to a Dynamic Port

Applications that do not use out-of-band signalling, or an IANA-assigned port still require both the client and server to agree the server port value to be used. This Section describes an optional method that allows an application to derive a default server port number from the Service Code. The returned value is in the dynamic port range [RFC4340]:

```
int s_port; /* server port */
s_port = (sc[0]<<7)^(sc[1]<<5)^(sc[2]<<3)^sc[3] | 0xC000;
if (s_port==0xFFFF) {s_port = 0xC000}
```


Where sc[] represents the four bytes of the Service Code, and sc[3] is the least significant byte, for example this function associates

SC:fdpz with the server port 64634.

This algorithm has the following properties:

- It identifies a default server port for each service.

- It seeks to assign different Service Codes to different ports, but does not guarantee an assignment is unique.

- It preserves the four bits of the final bytes of the Service Code, allowing mapping common series of Service Codes to adjacent ports, e.g. Foo1, and Foo2; and Fooa and Foob would be assigned adjacent ports.

- It avoids the port 0xFFFF, which is not accessible on all host platforms.

Applications and higher-layer protocols that have been assigned a Service Code (or use a Service Code from the unassigned private space) may use this method. It does not preclude other applications using the selected server port, since DCCP servers are differentiated by the Service Code value.

3. Use of the DCCP Service Code

The basic operation of Service Codes is as follows:

A client initiating a connection:

  . issues a DCCP-Request with a Service Code and chooses a destination (server) port number that is expected to be associated with the specified Service Code at the destination.

- A server that receives a DCCP-Request:

  . determines whether an available service matching the Service Code is supported for the specified destination server port. The session is associated with the Service Code and a corresponding server. A DCCP-Response is returned.

  . if the service is not available, the session is rejected and a DCCP-Reset packet is returned.

3.1. Setting Service Codes at the Client

A client application MUST associate every DCCP connection (and hence every DCCP active socket) with a single Service Code value [RFC4340]). This value is used in the corresponding DCCP-Request packet.

3.2. Using Service Codes in the Network

DCCP connections identified by the Service Code continue to use IP addresses and ports, although neither port number may be Published.

Port numbers and IP addresses are the traditional methods to identify a flow within an IP network. Middlebox [RFC3234] implementors therefore need to note that new DCCP connections are identified by the pair of Server Port and Service Code. This means that the IANA may allocate a server port to more than one application.

Network address and port translators, known collectively as NATs [RFC2663], may interpret DCCP ports [RFC2993] [ID.Behave-DCCP]. They may also interpret DCCP Service Codes. Interpreting DCCP Service Codes can reduce the need to correctly interpret port numbers, leading to new opportunities for network address and port translators. Although it is encouraged to associate specific delivery properties with the Service Code, e.g. to identify the real-time nature of a flow that claims to be using RTP, there is no guarantee that the actual connection data corresponds to the associated Service Code. A middlebox implementor may still use deep packet inspection, and other means, in an attempt to verify the content of a connection.

The use of the DCCP Service Code can potentially lead to interactions with other protocols that interpret or modify DCCP port numbers [RFC3234]. The following additional clarifications update the description provided in section 16 of RFC 4340:

- "A middlebox that intends to differentiate applications SHOULD test the Service Code in addition to the destination or source port of a DCCP-Request or DCCP-Response packet.

- A middlebox that does not modify the intended application (e.g. NATs [ID.Behave-DCCP] and Firewalls), MUST NOT change the Service Code.

- A middlebox MAY send a DCCP-Reset in response to a packet with a Service Code that is considered unsuitable."

3.3. Using Service Codes at the Server

A Service Code is used by a server that receives a DCCP-Request to associate a new DCCP connection with the corresponding application service. A number of options are presented for servers using passively listening sockets. Four cases can arise when two DCCP server applications listen on the same host:

- The simplest case arises when two servers are associated with different Service Codes and are bound to different server ports (section 3.3.1).

- Two servers may be associated with the same DCCP Service Code value, but be bound to different server ports (Section 3.3.1).

- Two servers could use different DCCP Service Code values, and be bound to the same server port (section 3.3.2).

- Two servers could attempt to use the same DCCP Service Code and bind to the same server port. A DCCP implementation MUST disallow this, since there is no way for the DCCP host to direct a new connection to the correct server application.

RFC 4340 (section 8.1.2) states that an implementation:

- MUST associate each active socket with exactly one Service Code on a specified server port.

In addition, section 8.1.2 also states:

- "Passive sockets MAY, at the implementation's discretion, be associated with more than one Service Code; this might let multiple applications, or multiple versions of the same application, listen on the same port, differentiated by Service Code."

This document updates this text in RFC 4340 by replacing this with the following:

- "An implementation SHOULD allow more than one Service Code to be associated with a passive server port, enabling multiple applications, or multiple versions of an application, to listen on the same port, differentiated by the associated Service Code."

It also adds:

- "An implementation SHOULD provide a method that informs a server of the Service Code value that was selected by an active connection."

3.3.1. Reception of a DCCP-Request

When a DCCP-Request is received, and the specified destination port is not bound to a server, the host MUST reject the connection by issuing a DCCP-Reset with Reset Code "Connection Refused". A host MAY also use the Reset Code "Too Busy" ([RFC4340], section 8.1.3).

When the requested destination port is bound to a server, the host MUST also verify that the server port is associated with the specified Service Code. Two cases can occur:

- If the receiving host is listening on a server port and the DCCP-Request uses a Service Code that is associated with the port, the host accepts the connection. Once connected, the server returns a copy of the Service Code in the DCCP-Response packet completing the initial handshake [RFC4340].

- If the server port is not associated with the requested Service Code, the server SHOULD reject the request by sending a DCCP-Reset packet with Reset Code 8, "Bad Service Code" ([RFC4340], Section 8.1.2), but MAY use the reason "Connection Refused".

A single application may wish to accept connections for more than one Service Code using the same server port. This may allow a server to offer more than the limit of 65,536 services determined by the size of the Port field. The upper limit is based solely on the number of unique connections between two hosts (i.e., 4,294,967,296).

After a connection has been accepted, the protocol control block is associated with a pair of ports and a pair of IP addresses and a single Service Code value.

3.3.2. Multiple Associations of a Service Code with Ports

RFC4340 states that a single passively opened (listening) port MAY be associated with multiple Service Codes, although an active (open) connection can only be associated with a single Service Code.

3.3.3. Automatically launching a Server

A host implementation may permit a service to be associated with a server port (or range of ports) that is not permanently running at the server. In this case, the arrival of a DCCP-Request may require a

method to associate a DCCP-Request with a server that handles the corresponding Service Code. This operation could resemble that of "inetd" [inetd].

As in the previous Section, when the specified Service Code is not associated with the specified server port, the connection MUST be aborted and a DCCP Reset message sent [RFC4340].

4. DCCP Benchmarking Services

A number of simple services are commonly supported by systems using TCP and UDP, this Section defines corresponding services for DCCP [RFC4340]. These services are useful for debugging DCCP implementations and deployment, and for benchmarking bidirectional DCCP connections. The IANA Section of this document allocates a corresponding set of code points for these services.

4.1. Echo

The operation of the DCCP echo service follows that specified for UDP [RFC862]: a server listens for DCCP connections; once a client has set up a connection, each data packet sent to the server will be copied (echoed) back to the client.

4.2. Daytime

The DCCP daytime service is operationally equivalent to the connection-based TCP daytime service [RFC867]: any data received is discarded by the server; and generates a response sent in a DCCP data packet containing the current time and date as an ASCII string; after which the connection is closed.

4.3. Character generator

The operation of the DCCP chargen service corresponds to the connection-based TCP chargen protocol [RFC864]: A server listens for incoming requests and, once a client has established a connection, continuously sends datagrams containing a random number (between 0 and 512, not exceeding the current DCCP Maximum Packet Size, MPS) of characters. The service terminates when the user either closes or aborts the connection. Congestion control is enforced using the mechanisms [RFC4340] and related documents.

If necessary, the receiver can enforce flow control on this service by using either or both of the Slow Receiver ([RFC4340], section 11.6) and Data Dropped ([RFC4340], section 11.7) DCCP options to signal the server to slow-down.

The chargen protocol provides a service that may be used for testing and measurement of bidirectional DCCP connectivity, as well as congestion control responsiveness. The datagram-based variant of chargen can be emulated with the DCCP ECHO service by changing the format of the datagrams sent by the client, hence these services complement each other.

4.4. Time service

The format of timestamps and the operation of the DCCP time service is equivalent to the TCP time protocol variant [RFC868]: a server listens for incoming connections; after a client has established a new connection, the server sends a 4-byte timestamp; whereupon the client closes the connection.

4.5. Generic PerfTest service

The PerfTest service specified by this document provides a generic service that may be used to benchmark and measure both unidirectional and bidirectional DCCP connections, as well as server and host DCCP stacks. These services are identified by the Service Code "XPER". This document does not specify a specific port number for this service.

The payload of DCCP packets associated with this service do not have a specified format. They are silently discarded by the receiver, and used only for gathering numerical performance data. Tools that have specific payload formats should register their own Service Code value

with IANA (e.g., section 4.6).

This Service Code is for benchmarking applications that transmit data in one direction only, with DCCP control traffic flowing in the opposite direction. A benchmarking application that expects data responses to the messages it sends would require a different Service Code. (This could result in different Middlebox treatment.)

4.6. PERF service

The PERF service specified by this document describes the service supported by the open-source iperf benchmarking program [iperf]. This may be used to benchmark and measure both unidirectional and bidirectional DCCP connections, as well as server and host DCCP stacks. This service is identified by a Service Code "PERF" and is associated with a well-known port number that currently coincides with the UDP port used by the iperf benchmarking program [iperf].

5. Security Considerations

This document discusses the usage of Service Codes. It does not describe new protocol functions. There are four areas of security that are important:

1. Server Port number reuse (section 5.1).

2. Interaction with NATs and firewalls (section 3.2 describes middlebox behaviour). Requirements relating to DCCP are described in [ID.Behave-DCCP].

3. Interpretation of DCCP Service Codes over-riding traditional use of reserved/Well Known port numbers (Section 5.2).

4. Interaction with IPsec and DTLS security (section 5.3).

5.1. Server Port number re-use

Service Codes are used in addition to ports when demultiplexing incoming connections. This changes the service model to be used by applications and middleboxes. The port-numbers registry already contains instances of multiple application registrations for a single

port number for TCP and UDP. These are relatively rare. Since the DCCP Service Code allows multiple applications to safely share the same port number, even on the same host, server port number reuse in DCCP may be more common than in TCP and UDP.

5.2. Association of applications with Service Codes

Care needs to be exercised when interpreting the mapping of a Service Code value to the corresponding service. The same service (application) may be accessed using more than one Service Code. Examples include the use of separate Service Codes for an application layered directly upon DCCP and one using DTLS transport over DCCP [RFC5238]. Other possibilities include the use of a private Service Code that maps to the same application as assigned to an IANA-defined Service Code value, or a single application that provides more than one service. Different versions of a service (application) may also be mapped to a corresponding set of Service Code values.

Processing of Service Codes may imply more processing than currently associated with incoming port numbers. Implementers need to guard against increasing opportunities for Denial of Service attack.

5.3. Interactions with IPsec

IPsec uses port numbers to perform access control in transport mode [RFC4301]. Security policies can define port-specific access control (PROTECT, BYPASS, DISCARD), as well as port-specific algorithms and keys. Similarly, firewall policies allow or block traffic based on port numbers.

Use of port numbers in IPsec selectors and firewalls may assume that the numbers correspond to Well Known services. It is useful to note that there is no such requirement; any service may run on any port, subject to mutual agreement between the endpoint hosts. Use of the Service Code may interfere with this assumption both within IPsec and in other firewall systems, but it does not add a new vulnerability. New implementations of IPsec and firewall systems may interpret the Service Code when implementing policy rules, but should not rely on either port numbers or Service Codes to indicate a specific service.

This is not an issue for IPsec because the entire DCCP header and payload are protected by all IPsec modes. None of the DCCP header is protected by application-layer security, e.g., DTLS [RFC5238], so again this is not an issue [RFC4347].

5.4. Security Considerations for Benchmarking Services

Services used for benchmarking and testing may also be used to generate traffic for other purposes. They can therefore pose an opportunity for a Denial of Service attack. Care needs to be exercised when enabling these services in an operational network. Appropriate rate-limits should be provided to mitigate these effects for servers provided for testing. In this respect, the security considerations are the same as those for other IETF-defined transport protocols.

6. IANA Considerations

This document does not update the IANA allocation procedures for the DCCP Port Number and DCCP Service Codes Registries as defined in RFC 4340.

6.1. IANA Assignments for Benchmarking Applications

A set of new services are defined in Section 4. Their corresponding IANA assignments are summarized in this Section.

This document notes that it is not required to supply an approved document (e.g. a published RFC) to support an application for a DCCP Service Code or port number value, although RFCs may be used to request Service Code values via the IANA Considerations Section. A specification is however required to allocate a Service Code that uses a combination of ASCII digits, uppercase letters, and character space, '-', '.', and '/') [RFC4340].

6.1.1. Port number values allocated by this document

IANA action is required to assign server ports for use by DCCP. This document requests allocation of the following code points from the IANA DCCP Port numbers registry:

>>>>>> IANA ACTION Please replace IANA THIS RFC, with the allocated RFC number. <<<

```
echo      7/dccp   Echo SC:ECHO
# IETF dccp WG, [IANA - THIS RFC]
daytime   13/dccp  DayTime    SC:DTIM
# IETF dccp WG, [IANA - THIS RFC]
chatgen   19/dccp  Chargen    SC:CHAR
# IETF dccp WG, [IANA - THIS RFC]
time      37/dccp  Timeserver SC:TIME
# IETF dccp WG, [IANA - THIS RFC]
perf      5001/dccp iPerf  SC:PERF
# IETF dccp WG, [IANA - THIS RFC]
```

6.1.2. Service Code values allocated by this document

This document solicits IANA action to allocate the following code points from the Service Code registry [IANA.SC]. The requested assignments are listed below and summarized in table 1. This set of Service Codes may be utilized for testing DCCP implementations and transmission paths.

>>>IANA Please confirm these allocations. >>>

Service Code (SC)	ASCII Code	Port	Description	Ref
1162037327 0x4543484f	ECHO	7	Echo service	[RFC862]
1146374477 0x4454494d	DTIM	13	Daytime server	[RFC867]

1128808786	CHAR	19	Character generator (chargen)	[RFC864]	
0x43484152					
1414090053	TIME	37	Timeserver	[RFC868]	
0x54494d45					
1346720326	PERF	5001	iPerf	[*]	
0x50455246					
1481655634	XPER	-	Generic Performance Service	[*]	
0x58504552					

Table 1: Allocation of Service Codes by this document.

Notes:
1) Port is the default port associated with this service.
2) * Reference is this document.

7. Acknowledgments

This work has been supported by the EC IST SatSix Project. Significant contributions to this document resulted from discussion with Joe Touch, and this is gratefully acknowledged. The author also thanks Ian McDonald, Fernando Gont, Eddie Kohler, and the DCCP WG for helpful comments on this topic, and Gerrit Renker for his help in determining DCCP behaviour and review of this document. Mark Handley provided significant input to the text on definition of Service Codes and their usage. He also contributed much of the material that has formed the historical background Section.

8. References

8.1. Normative References

[RFC1122] Braden, R. (ed.), "Requirements for Internet Hosts: Communication Layers, " STD 3, RFC 1122, Oct. 1989 (STANDARD).

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997 (BEST CURRENT PRACTICE).

[RFC4340] Kohler, E., M. Handley, S. Floyd, "Datagram Congestion Control Protocol (DCCP)", RFC 4340, Mar. 2006 (PROPOSED STANDARD).

[ID.Behave-DCCP] R. Denis-Courmont, "Network Address Translation (NAT) Behavioral Requirements for DCCP", IETF Work in Progress, draft-ietf-behave-dccp-02.txt.

8.2. Informative References

[IANA] Internet Assigned Numbers Authority, www.iana.org

[IANA.SC] IANA DCCP Service Code Registry http://www.iana.org/assignments/service-codes

[ID.Simul] G. Fairhurst, G. Renker, "DCCP Simultaneous-Open Technique to Facilitate NAT/Middlebox Traversal", IETF Work in Progress, draft-ietf-dccp-simul-open-01.txt.

[ID.RTP] C. Perkins, "RTP and the Datagram Congestion Control Protocol (DCCP)", IETF Work in Progress, draft-ietf-dccp-rtp-07.txt.

[ID.Rand] M. Larsen, F. Gont, "Port Randomization", IETF Work in Progress, draft-larsen-tsvwg-port-randomization-02.txt

[inetd] The extended inetd project, http://xinetd.org/

[iperf] http//dast.nlanr.net/Projects/Iperf/

[RFC768] Postel, J., "User Datagram Protocol", STD 6, RFC 768, August 1980.

[RFC793] Postel, J., "Transmission Control Protocol", STD 7, RFC 793, Sept. 1981 (STANDARD).

[RFC814] Clark, D., "NAME, ADDRESSES, PORTS, AND ROUTES", RFC 814, July 1982 (UNKNOWN).

[RFC862] Postel, J., "Echo Protocol", STD 20, RFC 862, May 1983.

[RFC864] Postel, J., "Character Generator Protocol", STD 22, RFC 864, May 1983.

[RFC867] Postel, J., "Daytime Protocol", STD 25, RFC 867, May 1983.

[RFC868] Postel, J. and K. Harrenstien, "Time Protocol", STD 26, RFC 868, May 1983.

[RFC2326] Schulzrinne, H., Rao, A., and R. Lanphier, "Real Time Streaming Protocol (RTSP)", RFC 2326, April 1998.

[RFC2663] Srisuresh, P. and M. Holdrege, "IP Network Address Translator (NAT) Terminology and Considerations", RFC 2663, August 1999.

[RFC2780] Bradner, S. and V. Paxson, "IANA Allocation Guidelines For Values In the Internet Protocol and Related Headers", BCP 37, RFC 2780, March 2000.

[RFC2782] Gulbrandsen, A., Vixie, P., and L. Esibov, "A DNS RR for specifying the location of services (DNS SRV)", RFC 2782, February 2000.

[RFC2993] Hain, T., "Architectural Implications of NAT", RFC 2993, November 2000.

[RFC3234] Carpenter, B. and S. Brim, "Middleboxes: Taxonomy and Issues", RFC 3234, February 2002.

[RFC3261] Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M., and E. Schooler, "SIP: Session Initiation Protocol", RFC 3261, June 2002.

[RFC3828] Larzon, L-A., Degermark, M., Pink, S., Jonsson, L-E., and G. Fairhurst, "The Lightweight User Datagram Protocol (UDP-Lite)", RFC 3828, July 2004.

[RFC4301] Kent, S. and K. Seo, "Security Architecture for the Internet Protocol", RFC 4301, December 2005.

[RFC4347] Rescorla, E. and N. Modadugu, "Datagram Transport Layer Security", RFC 4347, April 2006.

[RFC4566] Handley, M., Jacobson, V., and C. Perkins, "SDP: Session Description Protocol", RFC 4566, July 2006.

[RFC4960] Stewart, R., Ed., "Stream Control Transmission Protocol RFC 4960, September 2007.

[RFC5238] Phelan, T., "Datagram Transport Layer Security (DTLS) over the Datagram Congestion Control Protocol (DCCP)", RFC 5238, May 2008.

9. Author's Addresses

Godred (Gorry) Fairhurst,
School of Engineering,
University of Aberdeen,
Kings College,
Aberdeen, AB24 3UE,
UK
Email: gorry@erg.abdn.ac.uk
URL: http://www.erg.abdn.ac.uk/users/gorry

9.1. Intellectual Property Statement

9.2. Disclaimer of Validity

IETF TRUST AND THE INTERNET ENGINEERING TASK FORCE DISCLAIM ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE

ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

9.3. Copyright Statement

>>> RFC Editor please remove this Section prior to publication.

Change Log.

01 introduced:

- a replacement of the word *range* when referring to sets of dccp ports (they are not necessarily contiguous), noted by E. Kohler.

- Addition of some Service Codes in IANA Section.

02 introduced:

- add the use of profiles with DCCP, identified by Service Code, but not the use of protocol variants.

- further detail on implementation levels (more input would be good)

- added security consideration for traffic generators

- added ref to UDPL for completeness

- Corrected NiTs found by Gerrit Renker

++++++++++++++++++++++++++

WG 00 (first WG version)

This introduced revisions to make it a WG document.

- Corrected language and responded to many helpful comments from Fernando Gont and Ian McDonald.

- Added a test for which server behaviour is used.

- Added some speculative text on how to implement the SC.

- More input and discussion is requested from the WG.

- Added an informative appendix on host configuration.

- Merging of some Sections to remove repetition and clarify wording.

++++++++++++++++++++++++++

WG 01

Historical material was added.

Comments from the list have been included.

The concept of adding weak semantics to a SC=0 was removed. This was added at the request of implementers, with the aim of offering easier implementation on at least one target platform. It has been removed in this document because it weakens interoperability and complicates the Spec.

The proposal to allow several levels of support was introduced in previous drafts following suggestions from the WG, but was removed in this revision. The method was seen to introduce complexity, and resulted in complex interoperability scenarios.

Removed "test" method, this was no longer required.

Draft was reorganized to improve clarity and simplify concepts.

WG 02

Updated following comments from Eddie Kohler.

WG 03

Fixed NiTs and addressed issues marked in previous version.

Added 2 para at end of port Section saying how to use Well Known ports and that you do not need to register them.

WG 04

Cleaned English (removing duplication)

Checked text that updates RFC4340 (and remove duplicates).

Updated hash algorithm for SC->s_port

Updated to IANA Section.

Edits in response to feedback from Tom Phelan, et al.

WG-05:

Various Sections were updated following feedback from the list, some specific comments were:

Tom Phelan suggested clarification was needed for the usage of well-known ports in Section 1, and various other clarifications.

Eddie Kohler suggested reworking the midbox Section.

Eddie noted the hash function included the highest numbered port, which is not accessible on all OS.

There was also discussion about the proper server port range to be used with this method. After previous concerns that using registered ports could have some (unknown) side effect, use was recommended in the dynamic range. Text was added to this Section.

Discussions at IETF-71 lead to the idea to removing the IANA guidance

on maintaining the registries to a new document that defines the policy across the set of transport registries.

Eddie noted that port-reuse is likely to be more common with DCCP (security considerations).

Lars noted that rate-limiting benchmarking tools may be somewhat undesirable, and this related to services for testing.

The text recommending an update to the IANA procedures for ports and service codes has been moved to a TSV WG draft.

WG-06:

Updated the updating paragraphs to clarify the specific clauses of RFC 4340 are changed. Comments from Eddie and Colin.

Very minor editorial corrections.

WG-07:

Portname for Perf in registry changed to all lower case.

Replaced para 2 of intro and updated later parts of the introduction (feedback in LC from Eddie).

Added citation to the Behave WG Requirements for NATs (now in LC).

WG-08:

New text to address editorial corrections proposed by Alfred Hoenes.