RFC 2295

Network Working Group
Request for Comments: 2295
Category: Experimental

Transparent Content Negotiation

Status of this Memo

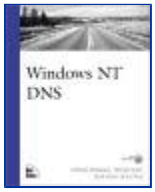
This memo defines an Experimental Protocol fo
community. It does not specify an Internet s
Discussion and suggestions for improvement ar
Distribution of this memo is unlimited.

Copyright Notice

ABSTRACT

HTTP allows web site authors to put multiple
information under a single URL. Transparent
an extensible negotiation mechanism, layered
automatically selecting the best version when
This enables the smooth deployment of new web
tags.

TABLE OF CONTENTS



1 Introduction

HTTP allows web site authors to put multiple information under a single URI. Each of thes `variant'. Transparent content negotiation i negotiation mechanism for automatically and e the best variant when a GET or HEAD request i the smooth deployment of new web data formats

This specification defines transparent conten extension on top of the HTTP/1.1 protocol [1] extension does not require use of HTTP/1.1: t negotiation can also be done if some or all o HTTP/1.0 [2] systems.

Transparent content negotiation is called `tr
makes all variants which exist inside the ori
outside parties.

Note: Some members of the IETF are currentl
of activities which are loosely related to
protocol. First, there is an effort to def
independent registry for feature tags. The
experimental protocol will be one of the cl
Second, some research is being done on cont
for other transport protocols (like interne
and on generalized negotiation systems for
protocols. At the time of writing, it is u
research will lead to results in the form o
system specifications. It is also unclear
future specifications can or will re-use el
experimental protocol.

1.1 Background

The addition of content negotiation to the we
been considered important since the early day
expected benefits of a sufficiently powerful
negotiation are

* smooth deployment of new data formats and
allow graceful evolution of the web

* eliminating the need to choose between a
multimedia homepage' and one which can be

* enabling good service to a wider range of
platforms (from low-end PDA's to high-end

* eliminating error-prone and cache-unfrien
User-Agent based negotiation

* enabling construction of sites without `c
version' links

* internationalization, and the ability to

content without a bias towards one langua

2 Terminology

The words "MUST", "MUST NOT", "SHOULD", "SHOU
this document are to be interpreted as descri

This specification uses the term `header' as
`header field in a request or response messag

2.1 Terms from HTTP/1.1

This specification mostly uses the terminolog
specification [1]. For the convenience of th
reproduces some key terminology definition fr

request
An HTTP request message.

response
An HTTP response message.

resource
A network data object or service that can b
Resources may be available in multiple repr
multiple languages, data formats, size, res
other ways.

content negotiation
The mechanism for selecting the appropriate
servicing a request.

client
A program that establishes connections for
requests.

user agent
The client which initiates a request. Thes
editors, spiders (web-traversing robots), o

server
An application program that accepts connect
requests by sending back responses. Any gi
capable of being both a client and a server
refers only to the role being performed by
particular connection, rather than to the p
general. Likewise, any server may act as a
gateway, or tunnel, switching behavior base
request.

origin server
The server on which a given resource reside

proxy
An intermediary program which acts as both
for the purpose of making requests on behal
Requests are serviced internally or by pass
possible translation, to other servers. A
both the client and server requirements of

age
The age of a response is the time since it
successfully validated with, the origin ser

fresh
A response is fresh if its age has not yet
lifetime.

2.2 New terms

transparently negotiable resource
A resource, identified by a single URI, whi
representations (variants) associated with
request on its URI, it allows selection of
using the transparent content negotiation m
transparently negotiable resource always ha
to it, which can be represented as an Alter
section 8.3).

variant list
A list containing variant descriptions, whi
transparently negotiable resource.

variant description
A machine-readable description of a variant in a variant list. A variant description c resource URI and various attributes which d the variant. Variant descriptions are defi

variant resource
A resource from which a variant of a negoti retrieved with a normal HTTP/1.x GET reques which does not use transparent content nego

neighboring variant
A variant resource is called a neighboring transparently negotiable HTTP resource if t a HTTP URL, and if the absolute URL of the its last slash equals the absolute URL of t up to its last slash, where equality is det comparison rules in section 3.2.3 of [1]. neighboring variant is important because of (section 14.2). Not all variants of a nego be neighboring variants. However, access t can be more highly optimized by the use of algorithms (section 7) and choice responses

remote variant selection algorithm
A standardized algorithm by which a server best variant on behalf of a negotiating use typically computes whether the Accept- head contain sufficient information to allow a c variant is the best variant. The use of a speed up the negotiation process.

list response
A list response returns the variant list of resource, but no variant data. It can be g does not want to, or is not allowed to, ret variant for the request. List responses ar 10.1.

choice response
A choice response returns a representation the request, and may also return the varian

the request, and may also return the varian
resource. It can be generated when the ser
information to be able to choose the best v
user agent, but may only be generated if th
neighboring variant. Choice responses are

adhoc response
An adhoc response can be sent by an origin
measure, to achieve compatibility with a no
client if this compatibility cannot be achi
or choice response. There are very little
contents of an adhoc response. Adhoc respo
section 10.3.

Accept- headers
The request headers: Accept, Accept-Charset
Accept-Features.

supports transparent content negotiation
From the viewpoint of an origin server or p
supports transparent content negotiation if
Negotiate header (section 8.4) which indica

server-side override
If a request on a transparently negotiated
client which supports transparent content n
server is said to perform a server-side ove
ignores the directives in the Negotiate req
uses a custom algorithm to choose an approp
server-side override can sometimes be used
client bugs. It could also be used by prot
of transparent content negotiation.

3 Notation

The version of BNF used in this document is t
of the nonterminals used are defined in [1].
underlying charset is US-ASCII.

One new BNF construct is added:

```
   1%rule

stands for one or more instances of "rule", s

   1%rule =  rule *( 1*LWS rule )

This specification also introduces

   number = 1*DIGIT

   short-float = 1*3DIGIT [ "." 0*3DIGIT ]
```


This specification uses the same conventions
1.2 of [1]) for defining the significance of
requirement.

4 Overview

This section gives an overview of transparent
It starts with a more general discussion of n
by HTTP.

4.1 Content negotiation

HTTP/1.1 allows web site authors to put multi
information under a single resource URI. Eac
called a `variant'. For example, a resource h
bind to three different variants of a paper:

1. HTML, English
2. HTML, French
3. Postscript, English

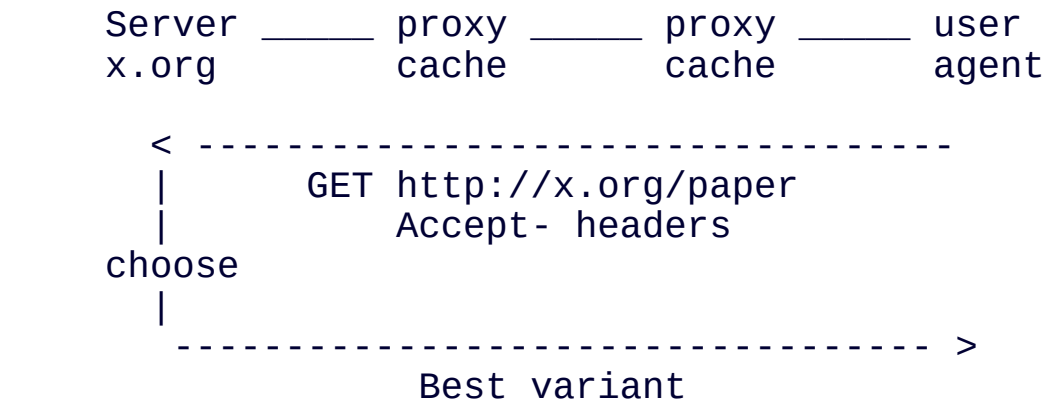
Content negotiation is the process by which t
selected if the resource is accessed. The se
matching the properties of the available vari
of the user agent and the preferences of the

It has always been possible under HTTP to hav

representations available for one resource, a
appropriate representation for each subsequen
HTTP/1.1 is the first version of HTTP which h
this in a cache-friendly way. These provisio
response header, entity tags, and the If-None

4.2 HTTP/1.0 style negotiation scheme

The HTTP/1.0 protocol elements allow for a ne
follows:



When the resource is accessed, the user agent
request) various Accept- headers which expres
capabilities and the user preferences. Then
these Accept- headers to choose the best vari
in the response.

The biggest problem with this scheme is that
For all but the most minimal user agents, Acc
all capabilities and preferences would be ver
them in every request would be hugely ineffic
because only a small fraction of the resource
multiple variants.

4.3 Transparent content negotiation scheme

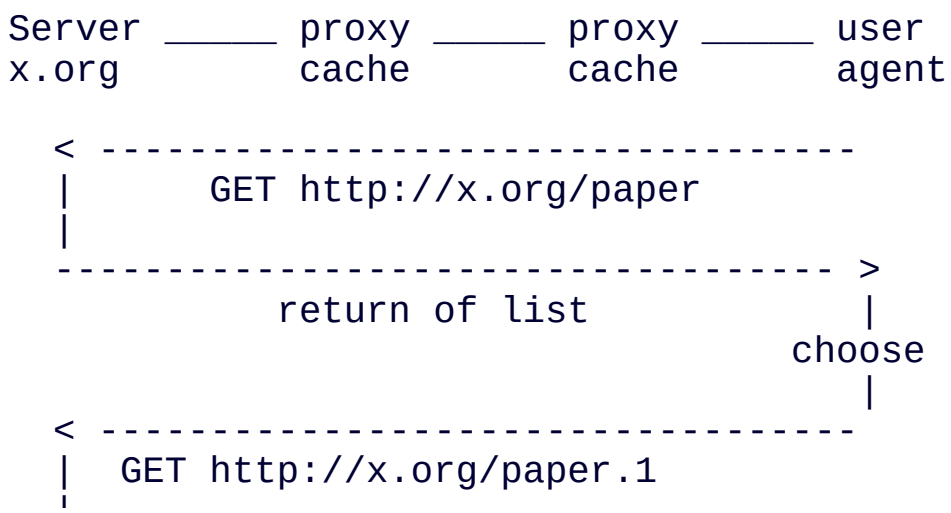
The transparent content negotiation scheme el
send huge Accept- headers, and nevertheless a
process that always yields either the best va
message indicating that user agent is not cap
of the available variants.

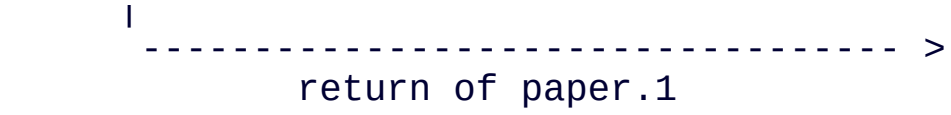
of the available variants.

Under the transparent content negotiation sch
list with the available variants and their pr
agent. An example of a list with three varia

```
{"paper.1" 0.9 {type text/html} {language
{"paper.2" 0.7 {type text/html} {language
{"paper.3" 1.0 {type application/postscrip
```

The syntax and semantics of the variant descr
covered in section 5. When the list is recei
choose the best variant and retrieve it. Gra
communication can be represented as follows:





The first response returning the list of vari
response'. The second response is a normal H
not contain special content negotiation relat
the user agent needs to know that the second
retrieves a variant. For the other parties i
second transaction is indistinguishable from
transaction.

With this scheme, information about capabilit
only used by the user agent itself. Therefor
information in large Accept- headers is unnec
do have a limited use in transparent content
sending of small Accept- headers can often sp
process. This is covered in section 4.4.

List responses are covered in section 10.1.
response in the above picture could be:

```
HTTP/1.1 300 Multiple Choices
Date: Tue, 11 Jun 1996 20:02:21 GMT
TCN: list
Alternates: {"paper.1" 0.9 {type text/html}
            {"paper.2" 0.7 {type text/html}
            {"paper.3" 1.0 {type applicatio
                {language en}}
Vary: negotiate, accept, accept-language
ETag: "blah;1234"
Cache-control: max-age=86400
Content-Type: text/html
Content-Length: 227
```

Multiple Choices:

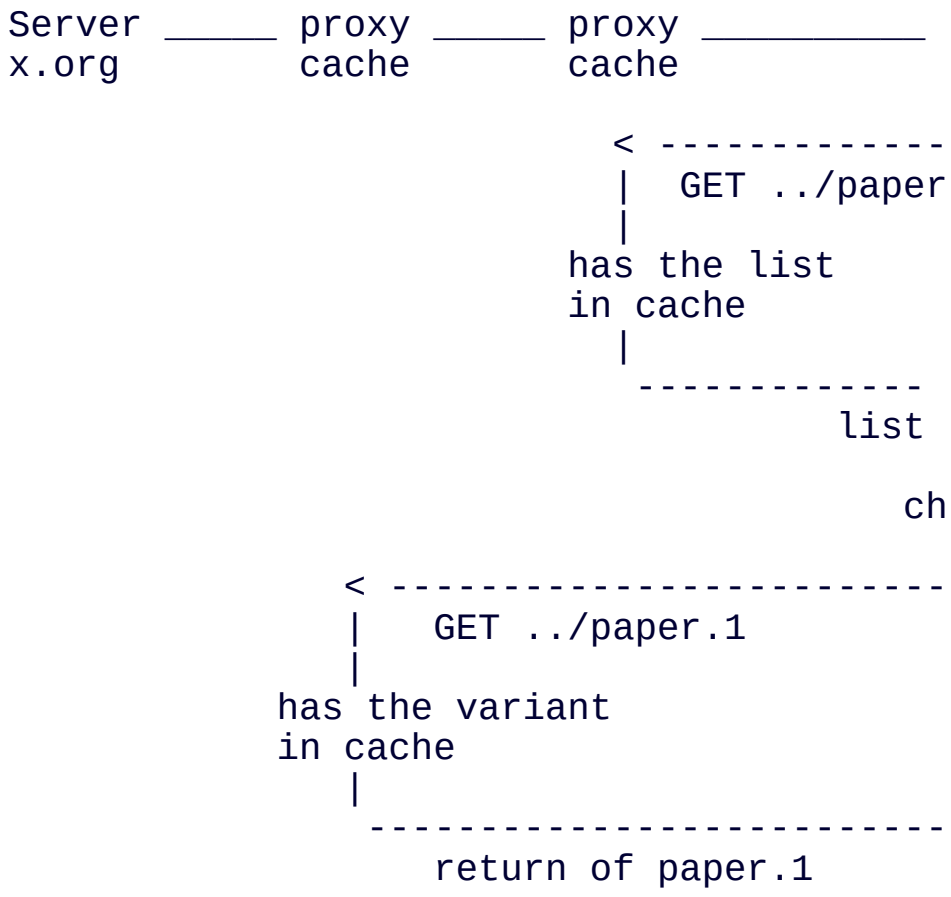
- HTML, English version
- HTML, French version
- Postscript, English version

The Alternates header in the response contain
Vary header is included to ensure correct cac
caches (see section 10.6). The ETag header a
revalidated by caches, the Cache-Control head
revalidation. The HTML entity included in th
user to select the best variant by hand if de

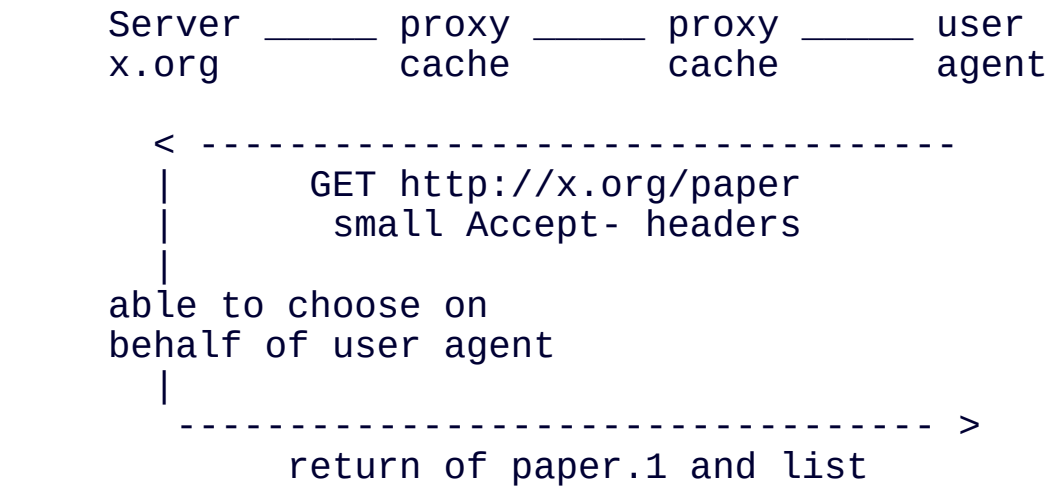
4.4 Optimizing the negotiation process

The basic transparent negotiation scheme invo
transactions: one to retrieve the list, and a
the chosen variant. There are however severa
in the data flow path of the basic scheme.

First, caching proxies can cache both variant
Such caching can reduce the communication ove
following example:



Second, the user agent can send small Accept- contain enough information to allow the serve variant and return it directly.



This choosing based on small Accept- headers variant selection algorithm'. Such an algori list and the Accept- headers as input. It th Accept- headers contain sufficient informatio the user agent, and if so, which variant is t best variant is a neighboring variant, it may with the variant list, in a choice response.

A server may only choose on behalf of a user transparent content negotiation if the user a the use of a particular remote variant select Negotiate request header. User agents with s variant selection algorithms may want to disa may want to allow it only when retrieving inl local algorithm of the user agent is superior areas of negotiation, it is possible to enabl for the easy areas only. More information ab variant selection algorithm can be found in [

Choice responses are covered in section 10.2. choice response in the above picture could be

```
HTTP/1.1 200 OK
Date: Tue, 11 Jun 1996 20:05:31 GMT
TCN: choice
Content-Type: text/html
Last-Modified: Mon, 10 Jun 1996 10:01:14 GM
Content-Length: 5327
Cache-control: max-age=604800
Content-Location: paper.1
Alternates: {"paper.1" 0.9 {type text/html}
```



```
{"paper.2" 0.7 {type text/html}
{"paper.3" 1.0 {type applicatio
    {language en}}
Etag: "gonkyyyy;1234"
Vary: negotiate, accept, accept-language
Expires: Thu, 01 Jan 1980 00:00:00 GMT
```

Multiple Choices for Web Stat

- Version with HTML tables
- Version without HTML tables
- Postscript version

```
blex
```

The Alternates header in the above script mus
line. The script always generates a list res
code, which ensures compatibility with non-ne

code, which ensures compatibility with non-ne
agents.

21.2 Direct support by HTTP servers

Sophisticated HTTP servers could make a trans
module available to content authors. Such a
a remote variant selection algorithm and an i
algorithm for generating choice responses (se
definition of interfaces to such modules is b
specification.

21.3 Web publishing tools

Web publishing tools could automatically gene
a document (for example the original TeX vers
tables, a HTML version without tables, and a
together with an appropriate variant list in
a HTTP server transparent negotiation module.
documents to be published as transparently ne

22 Appendix: Example of choice response construc

The following is an example of the constructi
by a proxy cache which supports HTTP/1.1 and
negotiation. The use of the HTTP/1.1 conditi
is also shown.

Assume that a user agent has cached a variant
"1234" for the negotiable resource http://x.o
that it has cached responses from two neighbo
entity tags "gonkyyyy" and W/"a;b". Assume t
cache entries are stale: they would need to b
user agent can use them. If http://x.org/pap
situation, the user agent could send the foll
proxy cache:

```
GET /paper HTTP/1.1
```

```
Host: x.org
User-Agent: WuxtaWeb/2.4
Negotiate: 1.0
Accept: text/html, application/postscript;q
Accept-Language: en
If-None-Match: "gonkyyyy;1234", W/"a;b;1234
```

Assume that the proxy cache has cached the sa
user agent, but that it has revalidated the v
ago, so that the list is still fresh for the
the proxy can run a remote variant selection
and the incoming request.

Assume that the remote algorithm is able to c
the best variant. The proxy can now construc
using the algorithm in section 10.2. In step
algorithm, the proxy can construct the follow
on the best variant, and send it to the origi

```
GET /paper.html.en HTTP/1.1
Host: x.org
User-Agent: WuxtaWeb/2.4
Negotiate: 1.0
Accept: text/html, application/postscript;q
Accept-Language: en
If-None-Match: "gonkyyyy", W/"a;b"
Via: 1.1 fred
```

On receipt of the response

```
HTTP/1.1 304 Not Modified
Date: Tue, 11 Jun 1996 20:05:31 GMT
Etag: "gonkyyyy"
```

from the origin server, the proxy can use its
paper.html.en cache entry to expand the respo
response:

```
HTTP/1.1 200 OK
Date: Tue, 11 Jun 1996 20:05:31 GMT
Content-Type: text/html
```

```
Last-Modified: Mon, 10 Jun 1996 10:01:14 GM
Content-Length: 5327
Cache-control: max-age=604800
Etag: "gonkyyyy"
Via: 1.1 fred
Age: 0
```