

IDR Working Group
Internet-Draft
Intended status: Standards Track
Expires: 29 September 2025

E. Chen
Palo Alto Networks
S. Sangli
Juniper Networks, Inc.
28 March 2025

Enhanced Dynamic Capability for BGP
draft-chen-idr-enhanced-dynamic-cap-00.txt

Abstract

This document addresses the limitations with the BGP Dynamic Capability specification by introducing additional protocol extensions and operational procedures so that a BGP capability that requires bi-directional capability advertisement can be revised dynamically.

Requirements Language

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 BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <https://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on 29 September 2025.

Copyright Notice

Copyright (c) 2025 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Revised BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Revised BSD License.

Table of Contents

1. Introduction	2
2. Enhanced Dynamic Capability	3
3. ENHANCED-CAPABILITY Message	3
4. Operational States for Capability Revision	6
4.1. For Local Capability Revision	6
4.2. For Remote Capability Revision	7
5. Operation	7
5.1. Interaction Between Sender and Receiver	8
6. When Does Capability Revision Take Effect	9
6.1. Uni-directional Advertisement	10
6.2. Bi-directional Advertisement	10
6.2.1. When Adding a Capability	10
6.2.2. When Deleting a Capability	11
7. Examples	11
7.1. Add Capability by One Speaker	11
7.2. Delete Capability by One Speaker	12
7.3. Add Capability Sequentially	13
7.4. Add Capability Concurrently	15
8. IANA Considerations	16
9. Security Considerations	17
10. Acknowledgments	17
11. References	17
11.1. Normative References	17
11.2. Informative References	17
Authors' Addresses	18

1. Introduction

The operation of certain BGP capabilities [RFC4271] [RFC5492] require bi-directional capability advertisement. The ADD-PATH Capability [RFC7911], ORF Capability [RFC5291], and Four-Octet AS Capability [RFC6793] are examples of such capabilities.

As discussed in the BGP Dynamic Capability specification [DYN-CAP], a capability that requires bi-directional capability advertisement can not be revised dynamically using the specified procedures due to lack of clear demarcation for the revised capability, in particular when the format of the UPDATE message is impacted.

This document addresses the limitations with the BGP Dynamic Capability specification by introducing additional protocol extensions and operational procedures so that a BGP capability that requires bi-directional capability advertisement can be revised dynamically. To avoid backward compatibility issues, a new BGP capability (termed "Enhanced Dynamic Capability") and a new BGP message type (termed "ENHANCED-CAPABILITY") are defined.

2. Enhanced Dynamic Capability

The Enhanced Dynamic Capability is a new BGP capability. The Capability Code for this capability is specified in the "IANA Considerations" section of this document. The Capability Value field consists of a list of capability codes (one-octet for each) that specify the capabilities that MAY be revised dynamically by the remote speaker.

As described in [RFC5492], a capability may have multiple instances defined. The Multiprotocol Extensions Capability specified in [RFC4760] is an example of such a capability. When including such a capability code in the Enhanced Dynamic Capability, a BGP speaker MUST make sure that all the capability instances recognized by the speaker are allowed to be revised dynamically by the remote speaker.

By advertising the Enhanced Dynamic Capability to a peer in the OPEN, a BGP speaker conveys to the peer that the speaker is capable of receiving and properly handling the ENHANCED-CAPABILITY message (as defined in the next Section) from the peer after the BGP session has been established.

The Enhanced Dynamic Capability itself is allowed to be revised dynamically.

3. ENHANCED-CAPABILITY Message

The ENHANCED-CAPABILITY Message is a new BGP message type. In addition to the fixed-size BGP header [RFC4271], the ENHANCED-CAPABILITY message contains the following fields:

```

+-----+
| Message Subtype (4 bits) |
+-----+
| Extra Parameters (4 bits) |
+-----+
| Reserved (7 bits) |
+-----+
| Action (1 bit) |
+-----+
| Capability Code (1 octet) |
+-----+
| Capability Length (2 octets) |
+-----+
| Capability Value (variable) |
+-----+

```

The "Message Subtype" field is an unsigned integer, and it specifies the subtype of the message. The following message subtypes are defined:

Subtype	Symbolic Name : Description
0	Init: for initiating a capability revision
1	Ack: for acknowledging a capability revision
2	AckConfirm: for confirming an Ack received
3	Nack: for rejecting a capability revision

For brevity, we use the subtype name plus "message" (e.g., Init message) to refer to the ENHANCED-CAPABILITY message with the specified subtype.

The "Extra Parameters" field is an unsigned integer, and is specific for each message subtype. Value 0 is reserved. This document defines the following values:

For the Ack and AckConfirm messages:

Value	Symbolic Description
1	Demarcation: capability revision applied

For the Nack message:

Value	Symbolic Description
1	Capability add: existing
2	Capability delete: non-existing
3	Capability add/delete: pending
4	Unexpected event: wrong FSM
5	Capability malformed

For other subtypes, the "Extra Parameters" field should be set to zero by the sender and ignored by the receiver.

The Reserved field should be set to zero by the sender and ignored by the receiver.

The Action field is 0 for advertising (or adding) a capability, and 1 for removing (or deleting) a capability.

Conceptually the triple (Capability Code, Capability Length, Capability Value) is the same as the one defined in [RFC5492], and it specifies a capability for which the "Action" shall be applied. The Capability Length field, though, is larger than the one specified in [RFC5492].

If multiple capability instances (as described in [RFC5492]) are defined for the capability code, then each capability instance SHALL be revised individually. The triple (Capability Code, Capability Length, Capability Value) in the ENHANCED-CAPABILITY message SHALL contain only one instance of the capability. If a BGP speaker does not recognize such a capability instance in a received ENHANCED-CAPABILITY message, it SHOULD log the case, but continue with the procedures of the capability revision.

If multiple capability instances (as described in [RFC5492]) are not explicitly defined for the capability code, but it has AFI/SAFI specific definitions (e.g., ADD-PATH), then the capability SHALL be treated as multi-instance (with each AFI/SAFI as a separate instance) for the purpose of dynamic capability revision in this document.

If multiple capability instances (as described in [RFC5492]) are not explicitly defined for the capability code, and it has no AFI/SAFI specific definitions (abbreviated as "single-instance capability" hereafter), then the "Action" specified applies to the whole capability identified by the capability code. Furthermore, if the "Action" is to remove a capability, then the Capability Length field SHOULD be set to zero by the sender and the Capability Value field MUST be ignored by the receiver even when the Capability Length field has a non-zero value.

If the "Action" is to remove a capability and the Capability Length field is zero, then the whole capability identified by the capability code is removed regardless whether multiple capability instances are defined for the capability code.

The triple (Capability Code, Capability Length, Capability Value) SHALL be used by a BGP speaker for matching an Ack, Nack, or AckConfirm message with the capability revision that a BGP speaker initiated previously.

The fields other than the "Message subtype" and "Extra Parameters" MUST remain unchanged from the original Init message during the progression of the capability revision.

4. Operational States for Capability Revision

A BGP speaker MUST maintain states about whether a capability has been advertised, or received during the lifetime of the BGP session. For a multi-instance capability, the states of the capability and its revision MUST be instance specific.

The following symbols are designated for that purpose:

L:Cap.True - Capability advertised
L:Cap.False - Capability not advertised

R:Cap.True - Capability received
R:Cap.False - Capability not received

Where the "L:" refers to the local speaker, and "R:" refers to the remote speaker.

During the dynamic revision of a capability, there are separate states, "Sending State", and "Receiving State", driving by the ENHANCED-CAPABILITY messages.

4.1. For Local Capability Revision

States for local capability revision:

```

L: Dyn. Oper. None/Add/Del
L: Send. None
L: Recv. None
L: Send. Init
L: Recv. Ack
L: Send. AckConfirm

```

State transitions:

```

L: Cap. True/False
L: Dyn. Oper. None
L: Send. None      ---> L: Send. Init ---> L: Recv. Ack
L: Recv. None      |
                    |
                    |
                    v
+----- L: SendAckConfirm -----+

```

4.2. For Remote Capability Revision

States for remote capability revision:

```

R: Dyn. Oper. None/Add/Del
R: Recv. None
R: Send. None
R: Recv. Init
R: Send. Ack
R: Recv. AckConfirm

```

State transitions:

```

R: Cap. True/False
R: Dyn. Oper. None
R: Send. None      ---> R: Recv. Init ---> R: Send. Ack
R: Recv. None      |
                    |
                    |
                    v
+----- R: RecvAckConfirm -----+

```

5. Operation

A BGP speaker MAY revise a capability using the ENHANCED-CAPABILITY message only when the capability is listed in the Enhanced Dynamic Capability of the received OPEN message. Furthermore, the speaker MUST NOT initiate a capability addition for a capability that has been advertised already. Also the speaker MUST NOT initiate a capability deletion for a capability that has not been advertised

before, such a capability revision SHALL be rejected by the receiver using a Nack message.

A BGP speaker MUST NOT initiate another revision of the the same capability (either for a single-instance capability, or for the same instance of a multi-instance capability) until the previous capability revision is complete. When a BGP speaker receives a revision for a capability that is already being revised, it SHALL send a Nack message rejecting the new revision. The "Extra Parameters" field SHOULD be populated accordingly.

When processing the ENHANCED-CAPABILITY message, if the Message Subtype is unrecognized, the speaker SHOULD log the case and ignore the message.

When processing the Init message, if the capability code or a capability instance (e.g., AFI/SAFI for ADD-PATH) is unrecognized, then the speaker SHOULD log the case but continue with the procedures for the capability revision.

A BGP speaker SHALL generate a Nack message with an appropriate "Extra Parameters" value when it detects an issue in processing an Init message.

When a BGP speaker receives a Nack message, it SHOULD log the error for further analysis. Depending on the severity of the issue determined by the "Extra Parameters" field, the speaker SHALL take the following actions:

- * For values 1, 2, 3: ignore the Nack, and abort the capability revision.
- * For other values: restart the bgp session with the desired capabilities in the OPEN message.

5.1. Interaction Between Sender and Receiver

For dynamically adding/deleting a capability listed in the Enhanced Dynamic Capability field of the OPEN message from a peer.

Time	Sender		Receiver	
	Event	State	Event	State
t0.1	Recv OPEN: Dynamic cap: cap code			
t0.2	Session established			
		L:Cap.True/False		R:Cap.True/False
		L:Dyn.Oper.None		R:Dyn.Oper.None
		L:Send.None		R:Recv.None
		L:Recv.None		R:Send.None
t1	Send Init	L:Send.Init L:Dyn.Oper.Add/Del		
t2			Recv Init	R:Recv.Init R:Dyn.Oper.Add/Del
t3			Send Ack	R:Send.Ack
t4	Recv Ack	L:Recv.Ack		
t5	Send AckConfirm and Re-Init	L:Send.AckConfirm L:Cap.True/False L:Dyn.Oper.None L:Recv.None L:Send.None		
t6			Recv AckConfirm and Re-init	R:Recv.AckConfirm R:Cap.True/False R:Dyn.Oper.None R:Recv.None R:Send.None

6. When Does Capability Revision Take Effect

A capability included in the Capabilities Optional Parameter [RFC5492] of the OPEN message, is considered complete on the sender (i.e., L:Cap.True), as well as on the receiver (i.e., R:Cap.True).

The dynamic revision of a capability is considered complete on the sender (i.e., L:Cap.True for "add", or L:Cap.False for "delete") after AckConfirm is sent, and on the receiver (i.e., R:Cap.True for "Add", or L:Cap.False for "delete") after the AckConfirm is received.

To support dynamic revision of the same capability by both speakers, the Demarcation field is introduced for the Ack and AckConfirm messages. The Demarcation field MUST be set when a speaker determines that the capability revision is ready to be applied, and the subsequent messages to the peer MUST be formatted with the capability revision applied.

If the Demarcation field is set in a received Ack or AckConfirm message, the receiver SHALL process subsequent messages (in particular the UPDATE message) from the peer based on the premise that the capability revision has been applied.

6.1. Uni-directional Advertisement

For a capability that does not require bi-directional advertisement, the Demarcation field in the Ack message SHALL be set.

6.2. Bi-directional Advertisement

For a capability that requires bi-directional advertisement, the criteria for determining when the capability revision would take effect depend on whether the capability has been advertised (including in the OPEN message), and whether the action is "add" or "delete", and the exchange of the Ack and AckConfirm messages.

6.2.1. When Adding a Capability

When a speaker sends an Ack message in response to an Init message from its neighbor, the Demarcation field of the Ack message SHALL be set under the following condition:

`R:Dyn.Oper.Add AND (L:Cap.True OR (L:Dyn.Oper.Add AND L:Send.Init))`

That is, if the local capability has been advertised, then the Demarcation field in the Ack message SHALL be set, and it SHALL operate with both the local capability and remote capability applied.

When a speaker sends an AckConfirm message in response to an Ack message from its neighbor, the Demarcation field of the AckConfirm message SHALL be set under the following condition:

`L:Dyn.Oper.Add AND (R:Cap.True OR (R:Dyn.Oper.Add AND R:Recv.Init))`

That is, if the remote capability has been received, then the Demarcation field in the AckConfirm message SHALL be set, and it SHALL operate with both the local capability and remote capability applied.

6.2.2. When Deleting a Capability

The Demarcation field SHALL be set in Ack, and AckConfirm in response to the Init or Ack messages, respectively, indicating the capability revision has been applied (i.e., disabled).

7. Examples

In this section several examples are provided for revising a capability that requires bi-directional capability advertisement, in particular, format changes to UPDATE messages are involved.

In the examples the term "Old Format" refers to the format of the UPDATE message without applying the capability. The term "New Format" refers to the format of the UPDATE message that has the capability applied. The IPv4-unicast Address Family for the ADD-PATH capability can be considered as a specific capability instance in these examples.

7.1. Add Capability by One Speaker

That is, one speaker advertises the capability in the OPEN message, and the other speaker revises the capability dynamically.

R1	R2
-----	-----
Send OPEN: Dynamic Cap N	Send OPEN: Cap N; Dynamic Cap N
~~~ Session Established ~~~	
L:Cap.False	R:Cap.False
R:Cap.True	L:Cap.True
L:Send.None	R:Recv.None
L:Recv.None	R:Send.None
R:Send.None	L:Recv.None
R:Recv.None	L:Send.None
~~~ UPDATE: Old Format ~~~	
Send UPDATE-1a: Old Format	
Send Init: L:Send.Init (Add)	
	Recv UPDATE-1a: Old Format
	Send UPDATE-2a: Old Format
	Recv Init: R:Recv.Init
	Send Ack: R:Send.Ack
	(Demarcation.on)
	Send UPDATE-2b: New Format

Recv UPDATE-2a: Old Format
 Send UPDATE-1b: Old Format

Recv Ack: L:Recv.Ack

Recv UPDATE-2b: New Format

Send AckConfirm: L:Send.AckConfirm
 (Demarcation.on)

Re-init: L:Cap.True
 L:Send.None
 L:Recv.None

Send UPDATE-1c: New Format

Recv UPDATE-1b: Old Format

Recv AckConfirm: R:Recv.AckConfirm
 Re-init: R:Cap.True
 R:Recv.None
 R:Send.None

Recv UPDATE-1c: New Format

7.2. Delete Capability by One Speaker

That is, both speakers advertise the capability in the OPEN messages, and then one speaker withdraws the capability dynamically.

R1	R2
-----	-----
Send OPEN: Cap N	Send OPEN: Cap N; Dynamic Cap N
~~~ Session Established ~~~	
L:Cap.True	L:Cap.True
R:Cap.True	R:Cap.True
L:Send.None	R:Recv.None
L:Recv.None	R:Send.None
R:Send.None	L:Recv.None
R:Recv.None	L:Send.None
~~~ UPDATE: New Format ~~~	
Send UPDATE-1a: New Format	
Send Init: L:Send.Init (Del)	
	Recv UPDATE-1a: New Format

Send UPDATE-2a: New Format

Recv Init: R:Recv.Init
 Send Ack: R:Send.Ack
 (Demarcation.on)

Send UPDATE-2b: Old Format

Recv UPDATE-2a: New Format
 Send UPDATE-1b: New Format

Recv Ack: L:Recv.Ack

Recv UPDATE-2b: Old Format

Send AckConfirm: L:Send.AckConfirm
 (Demarcation.on)

Re-init: L:Cap.False
 L:Send.None
 L:Recv.None

Send UPDATE-1c: Old Format

Recv UPDATE-1b: New Format

Recv AckConfirm: R:Recv.AckConfirm
 Re-init: R:Cap.False
 R:Recv.None
 R:Send.None

Recv UPDATE-1c: Old Format

7.3. Add Capability Sequentially

The capability is advertised by both speakers sequentially.

R1	R2
-----	-----
Send OPEN: Dynamic Cap N	Send OPEN: Dynamic Cap N
~~~ Session Established ~~~	
L:Cap.False	L:Cap.False
R:Cap.False	R:Cap.False
L:Send.None	R:Recv.None
L:Recv.None	R:Send.None
R:Send.None	L:Recv.None
R:Recv.None	L:Send.None

~~~ UPDATE: Old Format ~~~

Send UPDATE-1a: Old Format

Send Init: L:Send.Init (Add)

Recv UPDATE-1a: Old Format

Send UPDATE-2a: Old Format

Recv Init: R:Recv.Init

Send Ack: R:Send.Ack
(Demarcation.off)

Send UPDATE-2b: Old Format

Recv UPDATE-2a: Old Format

Send UPDATE-1b: Old Format

Recv Ack: L:Recv.Ack

Recv UPDATE-2b: Old Format

Send AckConfirm: L:Send.AckConfirm
(Demarcation.off)

Re-init: L:Cap.True
L:Send.None
L:Recv.None

Send UPDATE-1c: Old Format

Recv UPDATE-1b: Old Format

Recv AckConfirm: R:Recv.AckConfirm

Re-init: R:Cap.True
R:Recv.None
R:Send.None

Recv UPDATE-1c: Old Format

Send UPDATE-2c: Old Format

Send Init: L:Send.Init (Add)

Send UPDATE-2d: Old Format

Send UPDATE-1d: Old Format

Recv UPDATE-2c: Old Format

Recv Init: R:Recv.Init

Recv UPDATE-2c: Old Format

Send Ack: R:SendAck
(Demarcation.on)

Send UPDATE-1e: New Format

Recv UPDATE-1d: Old Format

Recv Ack: L:Recv.Ack
Send AckConfirm: L:Send.AckConfirm
(Demarcation.on)
Re-init: L:Cap.True
L:Send.None
L:Recv.NoNe

Recv UPDATE-1e: New Format
Send UPDATE-2c: New Format

Recv AckConfirm: R:Recv.AckConfirm
Re-init: R:Cap.True
R:Recv.None
R:Send.None

Recv UPDATE-2c: New Format

7.4. Add Capability Concurrently

The capability is advertised by both speakers at the same time.

| R1 | R2 |
|----------------------------------|----------------------------------|
| ----- | ----- |
| Send OPEN: Dynamic Cap N | Send OPEN: Dynamic Cap N |
| ~~~ Session Established ~~~ | |
| L:Cap.False | L:Cap.False |
| R:Cap.False | R:Cap.False |
| L:Dyn.Oper.None | L:Dyn.Oper.None |
| R:Dyn.Oper.None | R:Dyn.Oper.None |
| L:Send.None | R:Recv.None |
| L:Recv.None | R:Send.None |
| R:Send.None | L:Recv.None |
| R:Recv.None | L:Send.None |
| ~~~ UPDATE: Old Format ~~~ | |
| Send Init: L:Send.Init | Send Init: L:Send.Init |
| L:Dyn.Oper.Add | L:Dyn.Oper.Add |
| Recv Init: R:Recv.Init | Recv Init: R:Recv.Init |
| R:Dyn.Oper.Add | R:Dyn.Oper.Add |
| Send Ack: R:Send.Ack | Send Ack: R:Send.Ack |
| (Demarcation.on) | (Demarcation.on) |
| Recv Ack: L:Recv.Ack | Recv Ack: L:Recv.Ack |
| Tx AckConfirm: L:Send.AckConfirm | Tx AckConfirm: L:Send.AckConfirm |
| Re-init: L:Cap.True | Re-init: L:Cap.True |
| L:Dyn.Oper.None | L:Dyn.Oper.None |
| L:Send.None | L:Send.None |
| L:Recv.None | L:Recv.None |
| Rx AckConfirm: R:Recv.AckConfirm | Rx AckConfirm: R:Recv.AckConfirm |
| Re-init: R:Cap.True | Re-init: R:Cap.True |
| R:Dyn.Oper.None | R:Dyn.Oper.None |
| R:Recv.None | R:Recv.None |
| R:Send.None | R:Send.None |

8. IANA Considerations

This document introduces a BGP capability termed "Enhanced Dynamic Capability". The capability code needs to be assigned by IANA.

This document defines the ENHANCED-CAPABILITY message for BGP. The type code needs to be assigned by IANA.

9. Security Considerations

The extension proposed in this document does not change the underlying security or confidentiality issues inherent in the existing BGP [RFC4271].

10. Acknowledgments

TBD.

11. References

11.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC4271] Rekhter, Y., Ed., Li, T., Ed., and S. Hares, Ed., "A Border Gateway Protocol 4 (BGP-4)", RFC 4271, DOI 10.17487/RFC4271, January 2006, <<https://www.rfc-editor.org/info/rfc4271>>.
- [RFC4760] Bates, T., Chandra, R., Katz, D., and Y. Rekhter, "Multiprotocol Extensions for BGP-4", RFC 4760, DOI 10.17487/RFC4760, January 2007, <<https://www.rfc-editor.org/info/rfc4760>>.
- [RFC5492] Scudder, J. and R. Chandra, "Capabilities Advertisement with BGP-4", RFC 5492, DOI 10.17487/RFC5492, February 2009, <<https://www.rfc-editor.org/info/rfc5492>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.

11.2. Informative References

- [DYN-CAP] Chen, E. and S. Sangli, "Dynamic Capability for BGP", <draft-ietf-idr-dynamic-cap-17.txt, work in progress>.
- [RFC5291] Chen, E. and Y. Rekhter, "Outbound Route Filtering Capability for BGP-4", RFC 5291, DOI 10.17487/RFC5291, August 2008, <<https://www.rfc-editor.org/info/rfc5291>>.

- [RFC6793] Vohra, Q. and E. Chen, "BGP Support for Four-Octet Autonomous System (AS) Number Space", RFC 6793, DOI 10.17487/RFC6793, December 2012, <<https://www.rfc-editor.org/info/rfc6793>>.
- [RFC7911] Walton, D., Retana, A., Chen, E., and J. Scudder, "Advertisement of Multiple Paths in BGP", RFC 7911, DOI 10.17487/RFC7911, July 2016, <<https://www.rfc-editor.org/info/rfc7911>>.

Authors' Addresses

Enke Chen
Palo Alto Networks
3000 Tannery Way
Santa Clara, CA 95054
United States of America
Email: enchen@paloaltonetworks.com

Srihari Sangli
Juniper Networks, Inc.
Exora Business Park
Bangalore 560103
KA
India
Email: ssangli@juniper.net