

Systems and Protocol Aspects for Circumstellar Environments RG M. Piraux  
Internet-Draft  
Intended status: Informational  
Expires: 3 September 2026

Aerospacelab  
J. A. Fraire  
Inria / Saarland University  
2 March 2026

A code to describe satellite constellations  
draft-piraux-space-constellation-code-01

## Abstract

When considering a satellite constellation forming a non-terrestrial network, the characteristics of this constellation heavily influences the network topology it forms. To improve the analysis of such non-terrestrial networks across various tools developed by the network community, this document proposes a notation to describe common constellation patterns. In addition, this document may serve as an introduction to satellite constellations for IETF participants.

## About This Document

This note is to be removed before publishing as an RFC.

The latest revision of this draft can be found at <https://mpiraux.github.io/draft-piraux-space-constellation-code/draft-piraux-space-constellation-code.html>. Status information for this document may be found at <https://datatracker.ietf.org/doc/draft-piraux-space-constellation-code/>.

Discussion of this document takes place on the Systems and Protocol Aspects for Circumstellar Environments RG Research Group mailing list (<mailto:space@irtf.org>), which is archived at <https://mailarchive.ietf.org/arch/browse/space/>. Subscribe at <https://www.ietf.org/mailman/listinfo/space/>.

Source for this draft and an issue tracker can be found at <https://github.com/mpiraux/draft-piraux-space-constellation-code>.

## 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 3 September 2026.

## Copyright Notice

Copyright (c) 2026 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.

## Table of Contents

1. Introduction . . . . .	3
2. Conventions and Definitions . . . . .	4
3. Satellite constellations . . . . .	4
3.1. Walker constellations . . . . .	4
3.1.1. Walker Star . . . . .	5
3.1.2. Walker Delta . . . . .	6
4. Constellation code . . . . .	7
5. Examples of constellation codes . . . . .	9
6. Describing links in a shell . . . . .	9
6.1. Detailed specification . . . . .	10
6.1.1. Top-level keys . . . . .	10
6.1.2. Shell . . . . .	10
6.1.3. Link pattern . . . . .	11
6.1.4. Condition . . . . .	11
6.1.5. Expression . . . . .	11
7. Considerations for future versions of this document . . . . .	11
8. Security Considerations . . . . .	12
9. IANA Considerations . . . . .	12
10. References . . . . .	12
10.1. Normative References . . . . .	12
10.2. Informative References . . . . .	12
Appendix A. Changelog . . . . .	13
A.1. Since draft-piraux-space-constellation-code-00 . . . . .	13
Acknowledgments . . . . .	13
Authors' Addresses . . . . .	13

## 1. Introduction

The network topology of a satellite constellation is heavily influenced by its orbital characteristics. With recent technologies enabling Optical Inter-Satellite Links (OISL) between satellites, a network is formed by establishing links between neighbour satellites. The resulting topology can be dynamic as the distance between neighbour satellites changes throughout their orbital period. A common notation for the network community could improve the reproducibility of evaluations, measurements and simulations of satellite constellation networks.

The true position of a satellite is often represented using a Two-Line Element set (TLE). A TLE contains a number of fields describing the orbital elements at a given time of a given satellite. Combined with a simplified perturbation model, the TLE can be used to predict the future position and velocity of the satellite relatively accurately. However, when studying satellite constellations, TLEs may not be appropriate. First, they assume each satellite has a known absolute position, which is derived from the launch time and parameters which may not be known at the time of study. Second, they involve complex calculations given the chosen perturbation model which may not scale well to large-scale experiments. Third, TLEs are not sufficient to determine how the links are established within the constellation as they do not indicate its characteristics but only the position of its satellites.

The approach of this document is based on the mission parameters of a satellite constellation. Based on these parameters, the expected position of each satellite within the constellation can then be computed. While this approach does not capture the small discrepancies that can occur during the launch and operation of the satellites, we argue that it is sufficient in our context.

This version of the specification applies only to circular orbital shells. The rationale for this restriction is that circular orbits are the most common in current satellite constellations and simplify the code syntax. Elliptical orbits, such as those used in Molniya or Flower constellations, are outside the current scope but could be supported in a future extension of this document.

The notation defined in this document can also specify patterns for links within a shell of a constellation. Each pattern is repeated to establish the connectivity of a satellite with its neighbours within the shell. This is inspired by the works of network researchers on constellation network topology design [BhSi2019].

The rest of this document is organised as follows. Section 3 introduces two variants of the Walker pattern for orbital shells. These are used to define many of the existing satellite constellations. Section 4 defines the constellation code syntax using an ABNF grammar [RFC5234] and the code semantics. Section 5 contains examples of existing constellations defined using the constellation code. Section 6 augments the constellation code with a pattern notation to describe links within a shell. Finally, Section 7 concludes with considerations for future versions of this document.

## 2. Conventions and Definitions

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "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.

## 3. Satellite constellations

A constellation greatly improves the availability of a satellite service up to global or near-global coverage on Earth. From the user perspective, a constellation offers more guarantees that a satellite can be reached at all times. A constellation is composed of a set of orbital planes. Typically, several satellites are present on an orbital plane. They can be close together to perform formation flying or are equally spread within the plane. Orbital planes are distributed in a complementary manner, i.e., they share some properties (e.g. altitude and inclination) but differ in others (e.g. longitude of ascending node).

When all orbital planes of a constellation are circular orbits sharing the same altitude, they are said to constitute an orbital shell. Constellations often consist of a single orbital shell but more complex deployments can have several shells.

The rest of this section describes two common shells based on the Walker pattern.

### 3.1. Walker constellations

A Walker constellation consists of circular orbits sharing the same inclination. Two variants of the Walker pattern exist:

- \* Walker Star, where orbits are distributed over 180 degrees around the equator.

- \* Walker Delta, where orbits are distributed over 360 degrees around the equator.

### 3.1.1.1. Walker Star

Figure 1 is an illustration of a Walker Star constellation considering the Earth equator as horizontal in the Figure. The orbit trajectories are depicted by a dashed line, while satellites and their travel direction are indicated by arrow heads.

The orbits of a Walker Star constellation typically have an inclination close to 90 degrees with respect to the equator plane, though this is not a geometric constraint and other inclinations are possible. Given that they are distributed over 180 degrees around the equator plane, one half-sphere has satellites ascending from the south pole to the north pole while the other has them descending from north pole to south pole. This is depicted on the two sides of Figure 1. Over the south and north poles, all orbits are crossing paths before going over the other half-sphere.

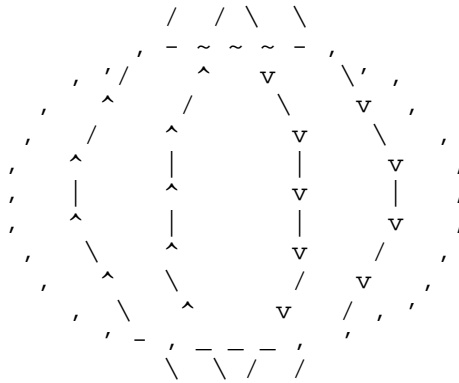


Figure 1: A Walker Star constellation

In a Walker Star constellation, a seam can be observed at the start and end of the orbit distribution around the equator plane. That is the first orbit (resp. last orbit) is next to the last orbit (resp. first orbit) going in the opposite direction of the sphere. It can be observed at the center of the Figure 1. The seam effect in Walker Star constellations may limit cross-plane ISL links at the seam boundary, though cross-plane links are still possible elsewhere; for instance, the Iridium constellation uses a Walker Star pattern with cross-plane ISLs. However, the Delta variant is often preferred for OISL-capable constellations due to the absence of the seam effect.

Figure 2 illustrates a part of a possible network topology for Walker Star constellations, with four orbital planes depicted vertically, each containing three satellites. In this example, links are only established in-plane, i.e., within the same orbit, though cross-plane links are also possible. Each orbit forms a ring, where the last satellite is connected to the first satellite.

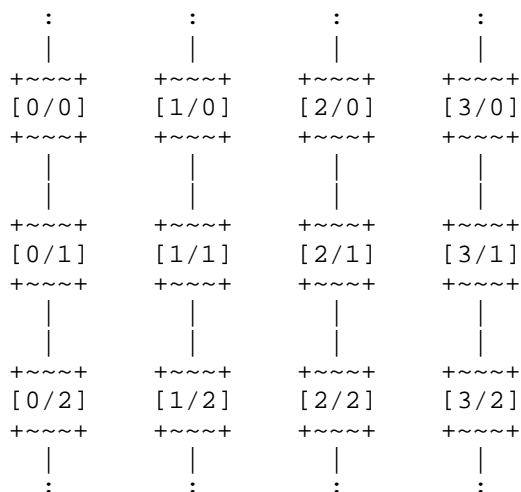


Figure 2: A Walker Star constellation network topology

### 3.1.2. Walker Delta

Figure 3 is an illustration of a Walker Delta constellation with only two orbits due to graphical constraints. The orbits of a Walker Delta constellation typically have an inclination ranging from 45 to 65 degrees with respect to the equator plane, though any inclination is geometrically valid. Combined with the altitude, the inclination directly limits the latitude coverage of a constellation, while Walker Star constellations have a complete latitude coverage.

Given that the orbits are distributed around the entire equator plane, there is no seam effect as in the Walker Star pattern. Instead, each orbit progresses in the same direction and cross paths twice with every other orbit. In this case, satellites can establish links with neighbouring orbits in addition to links within the same orbit.

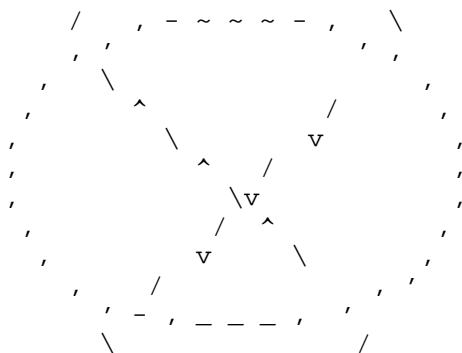


Figure 3: A Walker Delta constellation

Figure 4 illustrates a part of a possible network topology for Walker Delta constellations, with four orbital planes depicted vertically, each containing three satellites. Links are established in-plane and cross-plane, i.e., from one orbit to the other.

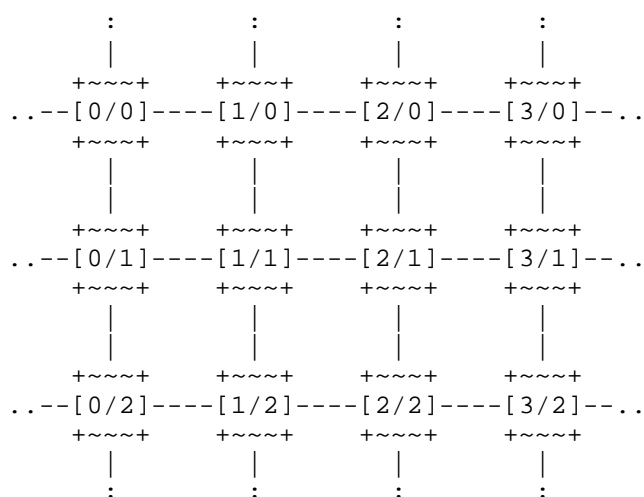


Figure 4: A Walker Delta constellation network topology

#### 4. Constellation code

Figure 5 defines the constellation code using an ABNF grammar [RFC5234]. The code can define a constellation with multiple shells. Each shell can follow a Walker Star or Walker Delta pattern.

```
constellation = shell [ "+" constellation ]
shell = walker ":" altitude ":" inclination ":" plane-params
shell =/ [ ":" mean-anomaly ]
walker = "D" / "S"
altitude = float
inclination = float
plane-params = no-sats "/" no-planes "/" phasing-factor
no-sats = int
no-planes = int
phasing-factor = int
mean-anomaly = float

int = 1*DIGIT
float = 1*DIGIT [ "." 1*DIGIT ]
DIGIT = %x30-39
```

Figure 5: ABNF Grammar for the constellation code

In addition to the grammar presented above defining the syntax of the code, a number of requirements on the semantics of the code are listed below.

- \* The altitude is expressed in kilometres with reference to the Earth's surface.
- \* The inclination is expressed in degrees and MUST be within the range of [0, 180] degrees. Inclinations greater than 90° represent retrograde orbits.
- \* The number of satellites must be evenly divisible by the number of planes.
- \* The phasing factor must be within the range [0, no-planes - 1]. It represents the relative offset between satellites in adjacent orbital planes. It determines how satellites in one plane are shifted in their orbital position compared to the satellites in the neighbouring plane, enabling optimal coverage patterns.
- \* The mean anomaly is expressed in degrees and MUST be within the range of [0, 360] degrees. It is optional and represents the orbital position of the first satellite in the first plane of the constellation. When absent it is considered equal to zero. The reference epoch for the mean anomaly is defined by the user's simulation environment or application context.



## 5. Examples of constellation codes

This section provides some examples of how the constellation code can be used to define existing satellite constellations sourced from public information. In some cases, when the phasing factor is not known, it is speculative.

Name	Description	Constellation code
Iridium	Walker Star, 780 km altitude, 86.4° inclination, 66 satellites, 6 planes	S:780:86.4:66/6/1
OneWeb	Walker Star, 1 200 km altitude, 87.9° inclination, 672 satellites, 12 planes	S:1200:87.9:672/12/11
Starlink (shell 1)	Walker Delta, 550 km altitude, 53° inclination, 1584 satellites, 72 planes	D:550:53:1584/72/39 [StFrHe2022]
GPS	Walker Delta, 20 180 km, 55° inclination, 24 satellites, 6 planes	D:20180:55:24/6/1

Table 1: Examples of constellation codes

## 6. Describing links in a shell

In this section, we extend the code notation with a complementary YAML format to specify the patterns of links within a shell. Each YAML document following the format defined in this I-D specifies a constellation that may be composed of several shells. An example of such a document is as follows:

```
version: draft-piraux-space-constellation-code-01
shells:
- code: D:1200:55:400/20/19
  link_patterns:
  - rank_offset: 1    # Establish an in-plane link towards the next sat.
  - plane_offset: 1   # Establish a cross-plane link in a staggered pattern
    conditions:       # e.g. when only three links are possible.
      - eq: [{mod: [rank, 2]}, {mod: [plane, 2]}] # rank % 2 == plane % 2
- code: S:1210:89:52/4/1
  link_patterns:
  - rank_offset: 1    # Establish an in-plane link towards the next sat.
```

Figure 6: Example of YAML document specifying a two-shell constellation

Figure 6 specifies a two-shell constellation. The first shell is a Walker Delta shell in which satellites have three links towards neighbours. The second one is a Walker Star pattern with two in-plane links per satellite.

These patterns are encoded through the `link_patterns` key. It contains a list of patterns with optional conditions. Each pattern can specify how to reach a neighbour given local plane and rank offsets to establish a bidirectional link. For instance, the first pattern of the first shell specifies that a link is formed with the next satellite in the same orbit.

For each pattern, a list of conditions can be expressed with the `conditions` key. These are evaluated for each satellite within the shell to determine whether the corresponding pattern should be applied to form a link. By applying each patterns to all satellites, the set of links within the constellation shell is established.

## 6.1. Detailed specification

### 6.1.1. Top-level keys

`version` Indicates the version of this I-D that the YAML document should be interpreted with.

`shells` A list of shells.

### 6.1.2. Shell

`code` The shell code following the specification in Section 4.

`link_patterns` A list of link patterns.

### 6.1.3. Link pattern

`rank_offset` An integer specifying the offset in rank to reach the neighbour for this link.

`plane_offset` An integer specifying the offset in plane to reach the neighbour for this link. When this offset causes the plane index to wrap around to the first plane, the rank index of the target satellite is adjusted according to the phasing factor of the shell.

Both types of offset can be included for a single pattern. When one is absent, it is considered to be equal to zero. In addition, they naturally wrap around at the boundaries of a shell.

`conditions` A list of conditions that must be met for this link to be added.

### 6.1.4. Condition

Each condition is a predicate on two expressions encoded as mapping. The predicate is the key while the two expressions form a sequence as the value. This version of the document only specifies the equality predicate indicated by the `eq` key.

### 6.1.5. Expression

It can be an integer, a context element or an operation on two expressions. Context elements are strings and include `rank` for the current rank index and `plane` for the current plane index of the satellite considered when evaluating a given link pattern.

An operation is encoded as mapping, with the key defining the type of operation and the value containing a sequence with the two expressions. This version of the document only specifies the modulo operation indicated by the `mod` key.

## 7. Considerations for future versions of this document

The code presented in this document does not consider the capabilities of satellites within a constellation to establish links. It focuses on defining the stable network topology that is expected for a constellation. Future versions of this document could consider means to define the capabilities of Optical Communication Terminals (OCTs) used to establish ISLs. This is complementary to the description of the network topology, which forms more of an intent, while capabilities define the space of possible links.

## 8. Security Considerations

As the code specified in this document is foreseen as a user input into software that performs simulations, evaluations and analysis of satellite constellations, implementers SHOULD consider validation and sanitisation measures.

## 9. IANA Considerations

This document has no IANA actions.

## 10. References

### 10.1. 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/rfc/rfc2119>>.
- [RFC5234] Crocker, D., Ed. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", STD 68, RFC 5234, DOI 10.17487/RFC5234, January 2008, <<https://www.rfc-editor.org/rfc/rfc5234>>.
- [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/rfc/rfc8174>>.

### 10.2. Informative References

- [BhSi2019] Bhattacharjee, D. and A. Singla, "Network topology design at 27,000 km/hour", ACM, Proceedings of the 15th International Conference on Emerging Networking Experiments And Technologies pp. 341-354, DOI 10.1145/3359989.3365407, December 2019, <<https://doi.org/10.1145/3359989.3365407>>.
- [StFrHe2022] Stock, G., Fraire, J., and H. Hermanns, "Distributed On-Demand Routing for LEO Mega-Constellations: A Starlink Case Study", IEEE, 2022 11th Advanced Satellite Multimedia Systems Conference and the 17th Signal Processing for Space Communications Workshop (ASMS/SPSC) pp. 1-8, DOI 10.1109/asms/spsc55670.2022.9914716, September 2022, <<https://doi.org/10.1109/asms/spsc55670.2022.9914716>>.

[TvdLCode] Tim van der Lee, "Constellation Code", July 2023,  
<<https://github.com/Tim024/ConstellationCode>>.

## Appendix A. Changelog

### A.1. Since draft-piraux-space-constellation-code-00

- \* Add YAML format to specify link patterns within shells of a constellation.
- \* Improvement of the text and examples.

## Acknowledgments

We thank Tim van der Lee for his work on a code [TvdLCode] that served as the basis for this document.

## Authors' Addresses

Maxime Piraux  
Aerospacelab  
Email: [maxime.piraux@aerospacelab.com](mailto:maxime.piraux@aerospacelab.com)

Juan A. Fraire  
Inria / Saarland University  
Email: [juan.fraire@inria.fr](mailto:juan.fraire@inria.fr)