

SDF modeling for Digital Twin

draft-ietf-asdf-digital-twin-05

September 2026

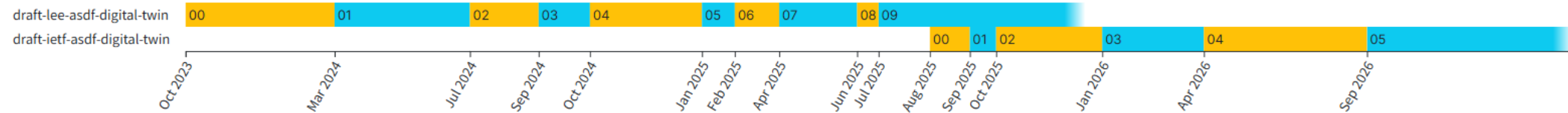
Hyunjeong Lee and Jungha Hong

ETRI

Changes since draft-04

Versions:

00 01 02 03 04 05



Section 5

Protocol Mapping draft alignment; BLE/Zigbee status clarified; Table 2 framed as examples

Section 5.4

Renamed to “QoS considerations”; MQTT QoS levels explicitly scoped as an example

Section 6

Renamed from “Examples” to “Digital twin system in various domains”; overview updated

Section 6.5

New Smart network system example based on network-device traffic information

Editorial

Terminology updated to RFC 9880, references refreshed, grammar and security wording improved

Table of contents: Version 04 → Version 05

Version 04

2. Terminology	3
3. SDF structure for digital twin	3
4. Motivation and design rationale	5
4.1. Introduction to sdfContext	5
4.2. Digital twin modeling using SDF elements	5
4.3. Relationship modeling	6
5. Protocol considerations for digital twin realization	8
5.1. Motivation	9
5.2. Supported protocol types	9
5.3. Protocol binding in SDF	10
5.4. QoS and Synchronization Semantics	10
5.5. Security and access considerations	11
5.6. Implementation guidelines	11
6. Examples of digital twin system	11
6.1. Overview	11
6.2. Marine system	11
6.3. Healthcare system	16
6.4. Smart building system	18
7. Requirements for implementing digital twin	20
8. Procedure for digital twin implementation	21
8.1. Overview	21
8.2. Procedure	21
9. Security Considerations	23
10. IANA Considerations	23
11. References	23
11.1. Normative References	23
11.2. Informative References	24
Acknowledgements	24
Contributors	24
Authors' Addresses	25



Version 05

2. Terminology	3
3. SDF structure for digital twin	3
4. Motivation and design rationale	5
4.1. Introduction to sdfContext	5
4.2. Digital twin modeling using SDF elements	5
4.3. Relationship modeling	6
5. Protocol considerations for digital twin realization	8
5.1. Motivation	9
5.2. Supported protocol types	9
5.3. Protocol binding in SDF	10
5.4. QoS considerations	11
5.5. Security and access considerations	11
5.6. Implementation guidelines	12
6. Digital twin system in various domains	12
6.1. Overview	12
6.2. Marine system	12
6.3. Healthcare system	17
6.4. Smart building system	19
6.5. Smart network system	21
7. Requirements for implementing digital twins	24
8. Procedure for digital twin implementation	24
8.1. Overview	24
8.2. Procedure	25
9. Security Considerations	26
10. IANA Considerations	26
11. References	26
11.1. Normative References	26
11.2. Informative References	28
Acknowledgements	28
Contributors	28
Authors' Addresses	28

5.4 title simplified

6.5 Smart network added

Section 7 typo fixed

References updated

Protocol considerations: clearer scope

What changed

- Protocol types are now explicitly described as examples for digital twin implementations
- Current SDF Protocol Mapping support is limited to BLE and Zigbee
- Table 2 keeps broader protocol candidates: MQTT, CoAP, BLE, Zigbee, HTTP/REST, WebSocket, NIPC

Key wording

Table 2 is intended to provide examples of protocol types that may be used for digital twin implementations; it does not imply that SDF protocol mappings have been defined for all of the listed protocols.

5. Protocol considerations for digital twin realization

5.1. Motivation

Digital twins require continuous and reliable communication with physical objects. To support synchronization, monitoring, control, and event notification, appropriate network protocols should be selected and semantically bound to modeled elements in SDF structures. This clause outlines the main protocol types, roles in digital twin operations, and guidelines for representing these bindings using [I-D.draft-ietf-asdf-sdf-protocol-mapping].

5.2. Supported protocol types

Digital twin applications can use different types of protocols depending on device performance, data volume, latency sensitivity, and network topology. The current SDF Protocol Mapping document defines mappings for BLE and Zigbee. Table 2 is intended to provide examples of protocol types that may be used for digital twin implementations; it does not imply that SDF protocol mappings have been defined for all of the listed protocols.

Protocol	Role in digital twin	Characteristics
MQTT	Sensor data publishing, event reporting	Lightweight, publish-subscribe, suitable for IoT
CoAP	REST-like access to constrained devices	UDP-based, compact, supports observe/notify
BLE	Local data exchange, control for wearables or embedded devices	Low energy, short range, uses characteristics/services
Zigbee	Sensor data access, event reporting, and device control for IoT devices	Uses endpoints, clusters, attributes, and commands for property, event, and action mappings
HTTP/REST	Enterprise integration, cloud API	Rich semantics, widely supported, heavier overhead
WebSocket	Bi-directional low-latency updates	State synchronization, real-time commands
NIPC	Standardized control of non-IP devices	Useful for industrial, air-gapped, or legacy systems

Table 2: Roles and characteristics for each protocol

Protocol binding and QoS: examples made more precise

5.3. Protocol binding in SDF

To model the way digital twins communicate with their physical objects, `sdfProtocolMap` can be defined within the `sdfProperty`, `sdfAction` or `sdfEvent` levels. Each protocol entry can specify a communication topic, path, security mechanism, QoS settings, and timing parameters as shown in [Figure 3](#).

```
{
  "sdfProperty": {
    "temperature": {
      "type": "number",
      "unit": "Cel",
      "sdfProtocolMap": {
        "ble": {
          "serviceID": "00001809-0000-1000-8000-00805f9b34fb",
          "characteristicID": "00002a1c-0000-1000-8000-00805f9b34fb"
        }
      }
    }
  }
}
```

Figure 3: An example of protocol binding

5.4. QoS considerations

Quality of service (QoS) settings define the reliability and frequency of data transfer between digital twins and physical objects. These settings are particularly important for telemetry (e.g., `sdfProperty` updates) and command response flows (e.g., `sdfAction` operations). For example, MQTT defines three QoS levels that can be selected according to the reliability requirements of digital twin data exchange, as shown in [Table 3](#).

QoS level	Meaning	Recommended Usage
0	At most once (best effort)	Periodic sensor data
1	At least once	State updates, events
2	Exactly once	Control command, AI action

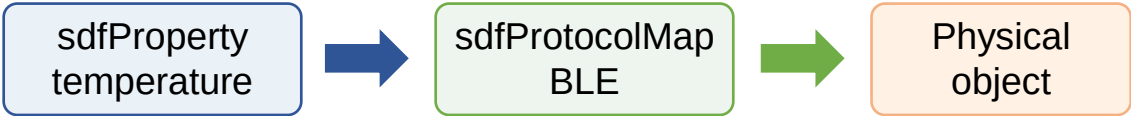
Table 3: Example MQTT QoS levels and recommended usage

5.5. Security and access considerations

When binding protocols, `sdfSecurityMap` can be used to include security parameters (e.g., authentication tokens, OSCORE for CoAP, BLE pairing status). Role-based access control for specific protocol endpoints is also recommended.

Updated modeling choices

- Figure 3 now uses BLE mapping fields: `serviceID` and `characteristicID`
- 5.4 title simplified to “QoS considerations”
- QoS table is now explicitly framed as an MQTT example



Section 6: Smart network system

Section 6 is reframed from an example list into domain coverage.

Marine system

Boat007 vessel model
with heaters and battery

Healthcare system

Patient monitor model
with ECG and infusion pump

Smart building

Lighting control with
occupancy context

Smart network

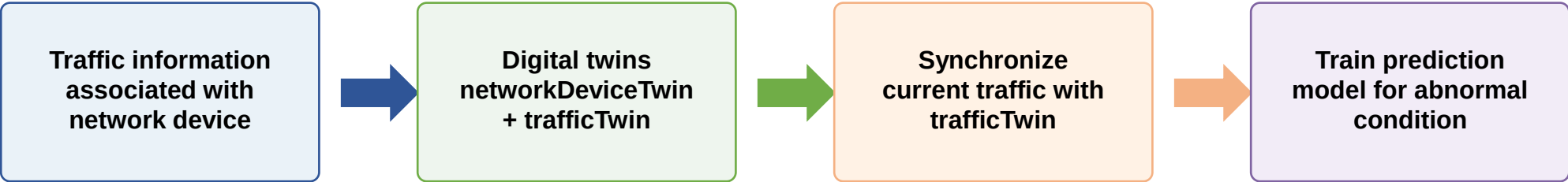
Network-device traffic
and prediction model

draft-04 “Examples of digital twin system”



draft-05 “Digital twin system in various domains”

New Section 6.5: Smart network system



Attribute	SDF element	Properties
Smart network system	sdfThing	smartNetwork001 including networkDeviceTwin and trafficTwin
networkDeviceTwin	sdfObject	device ID, status, currentTraffic, utilization, latency
trafficTwin	sdfObject / sdfData	currentTraffic, trafficHistory, trafficPattern
Current traffic	sdfProperty	throughput, packetRate, utilization, delay
Twin association	sdfRelation	networkDeviceTwin associated with trafficTwin
Synchronization	sdfRelation / sdfProperty	currentTraffic synchronized with trafficTwin state
Prediction result	sdfProperty / sdfEvent	anomalyScore, abnormalConditionPredicted

Table 7: Components and SDF elements of a smart network system

```

{
  "sdfThing": {
    "smartNetwork001": {
      "sdfObject": {
        "networkDeviceTwin": {
          "sdfProperty": {
            "currentTraffic": { "type": "number", "unit": "Mbit/s" },
            "utilization": { "type": "number", "unit": "percent" }
          }
        },
        "trafficTwin": {
          "sdfData": {
            "trafficHistory": { "type": "array" },
            "trafficPattern": { "type": "string" }
          },
          "sdfProperty": {
            "synchronizedCurrentTraffic": { "type": "number", "unit": "Mbit/s" }
          }
        }
      },
      "sdfRelation": {
        "trafficAssociation": {
          "source": "#/sdfObject/networkDeviceTwin",
          "target": "#/sdfObject/trafficTwin",
          "relType": "associatedWith"
        },
        "trafficSynchronization": {
          "source": "#/sdfObject/networkDeviceTwin/sdfProperty/currentTraffic",
          "target": "#/sdfObject/trafficTwin/sdfProperty/synchronizedCurrentTraffic",
          "relType": "synchronizedWith"
        }
      },
      "sdfProperty": {
        "anomalyScore": { "type": "number" }
      },
      "sdfEvent": {
        "abnormalConditionPredicted": {
          "description": "Raised when the trained model predicts an abnormal condition"
        }
      }
    }
  }
}

```

Figure 7: An example of smart network system

Editorial and reference updates

Area	Update
Terminology	SDF reference updated from draft to RFC 9880
References	Protocol Mapping reference updated to draft-ietf-asdf-sdf-protocol-mapping-11
Security	Authentication / authorization wording tightened
Procedures	Interworking and configuration bullets clarified

Impact The draft now reads less like an example collection and more like a structured modeling guide with protocol scope, domain examples, and lifecycle guidance aligned to current SDF references.

Next steps

- Consider AI-based components
 - “Digital Twin + AI Agent = Active system”
 - inference
 - decision
 - autonomous control

Thank you!

Q&A