FROZEN

Overall SPAN-AI Architecture v3 and

PROPOSAL FOR A UNIVERSAL CONTENT DISTRIBUTION NETWORK

BASED ON THE SPAN-AI-HARD SYSTEM

Background and Introduction

GT Systems, an Australian technology, media and telecom's company, has been developing media distribution systems for over a decade. We have been researching the problem of distributing video via the Internet for over two decades. That work began with the CSIRO and NICTA, Australia's leading R&D organisations. In the mid to late 2000s, Dr. Terry Percival, (head of research at CSIRO and NICTA, "father" of WiFi and now an adviser to GT Systems) commissioned some research work on "Peer Assist" networking, based on visioning workshops run by Rhett Sampson, the founder of GT Systems. That work combined BitTorrent peer to peer protocols with a central http server and obtained very encouraging results.

In 2008/9 GT Systems brought CD and DVD Manufacturing on Demand (MoD) technology to Australia. This developed close working relationships with all major Hollywood studios, Intel, Microsoft and retailers. In 2012, GT systems began work on developing a network technology to solve the "spinning wheel of death" (buffering and quality) problem in delivering movies via the Internet. Peer Assist was influential in the early days of that work but, ultimately, we saw the need to develop a next generation system that did not depend on BitTorrent.

In November 2014, GT Systems applied for an Australian and then international PCT patents for a "Media Distribution & Management System & Apparatus" based on our Secure Peer Assist (SPA) intelligent distributed storage network and routing criteria. In 2018 we met Jonathan Holt who became our adviser and helped us integrate the Protocol Labs IPFS stack into our SPA system. It was a perfect fit. Our US PCT patent was granted in May 2020. The same month we applied for an Australian patent and are in the process of applying for a US CIP patent extension for our ultimate goal of a widely applicable, universally operable, highly scalable and efficient, AI-driven system for optimisation, management and operation of a Universal Content Delivery Network (UCDN) incorporating our evolved Secure Peer Assist Network architecture (SPAN-AI). Our intent in filing that patent was to provide an equitable basis for discussion of how we disseminate the technology.

Relationship with Protocol Labs, IPFS and Filecoin

From our first encounter with IPFS it was clear that this was the distributed file system and protocol stack we were looking for and had intended to write to sit "underneath" our SPA media distribution architecture. Filecoin was the perfect implementation of our "frequent sharer points" system. We implemented both in our direct to market MVP app, Blust (see www.blust.tv).

In January 2019 we became aware of Protocol Labs R&D RFPs to scale routing/DHT and pub/sub. It became clear that our SPAN-AI technology was an excellent candidate solution. We formed a project team and designed a solution based on our own work, Protocol Labs work, their excellent problem statements and survey of the state of the art. In that design process we combined our SPAN-AI technology with IPFS, Name Resolution Networking, Name Data Networking and Information Centric Networking, along with several new innovations, to form an architecture and design for a Universal

Content Distribution Network (UCDN) incorporating our AI Hybrid Adaptive Routing Design, AI-HARD.

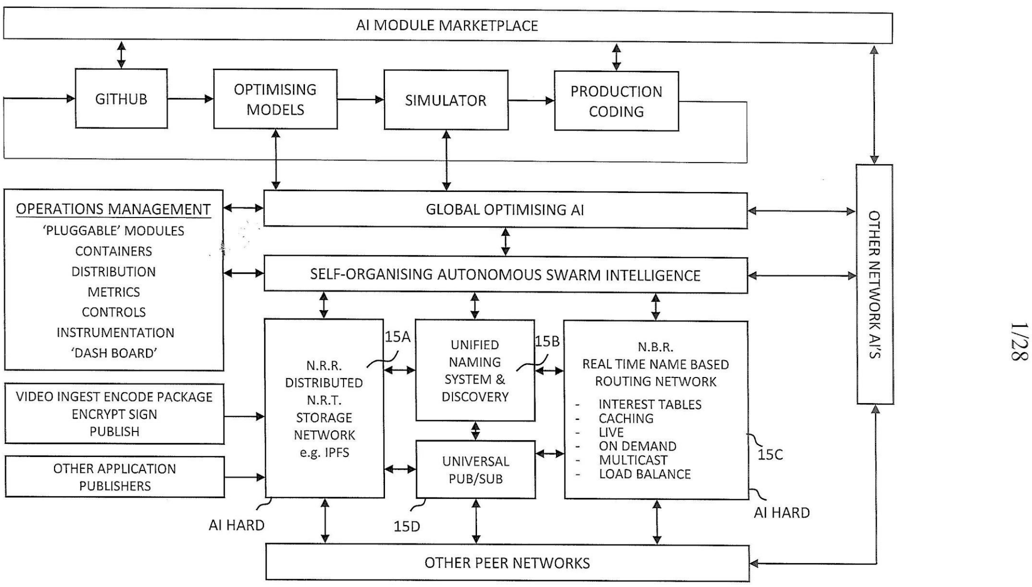
UCDN Overview

The UCDN incorporates AI-driven Secure Peer-Assisted Networking (SPAN-AI), an evolution of Secure Peer Assist (SPA). SPAN-AI is a hybrid adaptive networking technology that provides global, scalable, secure, distributed content storage, computation, and delivery for any application and network environment. SPAN-AI recognizes the limitations of existing technologies, only suitable for specific applications at non-global scale, and uses an AI-driven hybrid routing approach to improve and adaptively combine best-fit features of existing solutions under a unified, secure, content-addressable architecture. We call this a Universal Content Delivery Network or UCDN.

The objective of UCDN is to create a global heterogeneous network of inter-operable peer networks, thereby eliminating the problems associated to date with the "network of networks" approach. UCDN does that via open standards, interfaces, protocols and methods enabling any network to inter-operate with any other at both the AI and routing level.

UCDN Architecture:

The UCDN architecture is essentially the SPAN-AI architecture incorporating open standards, interfaces, protocols and methods enabling any network to inter-operate with any other at both the AI and routing level. This includes networks based on SPAN-AI architecture as well as existing TCP/IP and other network protocols and data models such as IPLD and blockchain.



UNIVERSAL CONTENT DISTRIBUTION NETWORK (UCDN)

Fig. 1

SPAN-AI Architecture

SPAN-AI is a heterogeneous, hierarchical network design based on 5 key systems: unified naming; universal discovery (together UND); hybrid adaptive routing; scalable publish/subscribe (pub/sub); and embedded security; all securely integrated and jointly optimized via a hierarchical, pluggable AI framework with an associated simulation, training, and development pipeline that embeds AI agents with varying degrees of awareness and optimization capabilities at peer, edge, and core network levels.

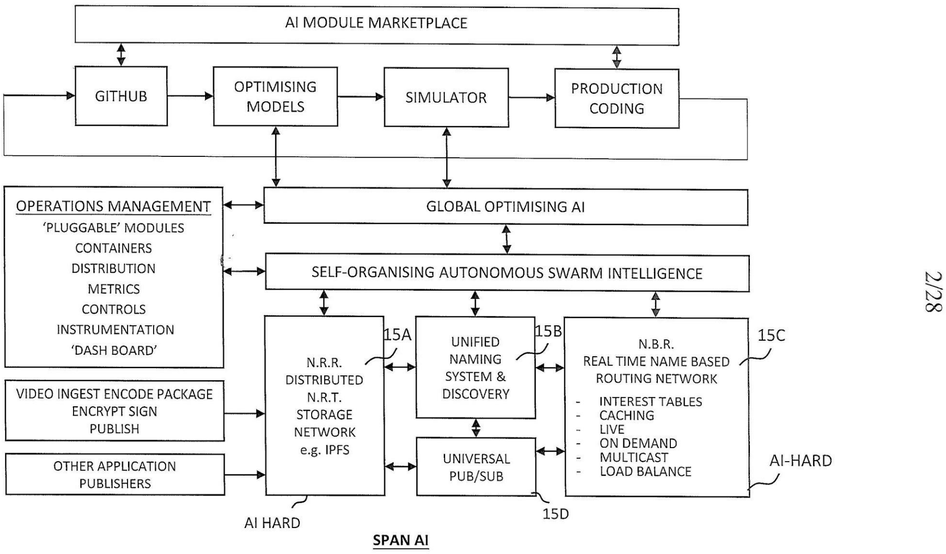


Fig. 1A

SPAN-AI Detailed Design

SPAN-AI uses a Unified Naming System that i) maps mutable human readable names (e.g., domain names, content names) to immutable self-certifying content identifiers (CIDs), and ii) enables routing CIDs through both name-resolution and name based routing subsystems, by iii) prepending a name prefix to each CID.

1. Unified Naming System
    a. SPAN-AI is content-addressable
    b. Content items or blocks are identified via immutable self-certifying content identifiers (CIDs), as in IPFS
    c. A global, distributed naming directory service is used to map mutable human-readable names/links to immutable CIDs
        i. Initially, IPNS and/or DNSLink will be used
        ii. Extensions include the use of NDNS

d. CIDs are then resolved (CID-provider mapping and provider-requester path formation) via a hybrid adaptive routing system (system 3)
    i. Name-resolution based routing, i.e., querying a (multi-level) DHT
    ii. Name based routing, i.e., hop-by-hop forwarding of an interest packet with a prefix (e.g., SPAN/<CID>)
e. Extensions include hierarchical names and name-based routing for name-CID mapping
f. SPAN-AI intelligence determines where to host distributed naming services (see SPAN-AI Intelligence section)
g. SPAN-AI pub/sub system is used for scalable, fast dissemination of naming updates (see Scalable Pub/Sub system)

SPAN-AI employs an AI-driven universal discovery system, whose core component, **Ambient Intelligence Rendezvous (AmI-Rendezvous)**, provides smart discovery, configuration, and self-organization services.

2. Universal Discovery System (AmI-Rendezvous)
    a. Provides smart discovery, configuration, and self-healing services
        i. Bootstrap nodes, resources
        ii. Discover peers, services
        iii. Maintain DHT and pub/sub overlays
    b. Combines peer-level self-healing intelligence and edge-level smart discovery (see AmI-Rendezvous operation in SPAN-AI intelligence section)
    c. SPAN-AI intelligence determines where to host distributed AmI-Rendezvous services (see SPAN-AI intelligence section)
        i. AmI-Rendezvous is ideally co-hosted with edge-level naming and intelligence services
        ii. Peers register on initialisation after performing mDNS and DHT discovery
        iii. Data partitioning and service placement may be guided by naming

SPAN-AI addresses routing at scale via an **AI-driven Hybrid Adaptive Routing Design (AI-HARD)**, composed of 2 subsystems, aimed at combining the benefits of name-resolution-based routing (NRR) for scalable, available, accessible distributed storage, and the advantages of name-based routing (NBR) for fast, reliable content delivery. AI-HARD includes storage and delivery markets.

3. AI-driven Hybrid Adaptive Routing Design (AI-HARD)

3.1 Storage-centric routing subsystem
    a. Main goal is persistent data availability (all content should be reachable) and relatively fast content access (< 1 sec)
    b. **Name-resolution-based routing (NRR)** via a parametrized **multi-level DHT** incorporating
        i. Learned information about user demands (e.g., content popularity, delivery deadlines) and network/topological structure (e.g., hop-distance, latency, load) used to create multiple, limited-size, fast-lookup layers
        ii. Topological layers
            1. Each DHT only involves topologically nearby nodes

2. Local, regional, national DHTs
   iii. Topic layers
      1. Each DHT only involves content related to a given topic
   iv. Heterogeneous layers
      1. Each DHT involves nodes that are both nearby and share similar interests

c. **Intelligent content placement** with adaptive replication level
   i. Content is replicated according to learned interest/popularity and network connectivity/stability (more replication under high churn/instability)
   ii. Replication level is optimized to guarantee lookup+delivery latency requirements of "storage-centric" applications. Additional in-network caching is provided for delivery-centric applications (see subsystem 3.2)
d. Multi-level DHT structure and associated parameters (layers, participant nodes, bucket size, concurrency factor), as well as content replication, are dynamically adjusted based on AI-driven optimization and distributed control algorithms (see AI-HARD operation in SPAN-AI Intelligence)
e. Multi-level DHT and smart content replication solutions allow maximizing the number of queries resolved locally in order to provide efficient, scalable, persistent content access
f. Content-level (as opposed to chunk-level) forwarding state to further improve scalability
g. Integrated with name-resolution-based subsystem via **common name directory service** (system 1)

3.2 Delivery-centric routing subsystem

a. Main goal is fast content delivery (< 100 ms)
b. **Name-based routing (NBR)** for fast lookup and delivery (e.g., NDN)
   i. Dataplane-aware symmetric Interest-Data packet forwarding
   ii. In-network caching
   iii. Native multicast and mobility support
   iv. In-network load balancing
c. Integrated with NRR subsystem via **common name directory service** (system 1)
d. Only used for applications with real-time requirements (e.g., live video streaming)
   i. Reserved to applications that can benefit from faster and more efficient application aware name based routing at the network level
   ii. Significantly reduces forwarding state (NRR subsystem handles apps with non real-time requirements)
   iii. Allows keeping chunk-level forwarding state to exploit path diversity and further speed-up content delivery

3.3 Market Enablement

a. Storage-centric

    i. Publishers may choose and pay for suitable storage metrics (reliability, duplication, dispersion, persistence, etc.) in a market such as Filecoin, Storj etc.
    ii. SPAN-AI supports multiple storage markets and technology platforms and unites them into a Unified Content Storage and Delivery Network. This may include storage markets supported by platforms such as blockchain.
b. Delivery-centric
    i. Publishers may choose and pay for suitable delivery metrics (resolution, bit rate, delay, etc.) in a market similar to Filecoin
    ii. Distribution providers (telco's, ISPs, CDNs, etc) may bid for delivery in the same market or rely on SPAN-AI and AI-HARD to choose the most efficient path, thereby incentivising efficiency
    iii. Consumers may choose which distributor or distributors they wish to use if, for example, they have come to an arrangement with any distributor. Consumers are free to choose if and who they make arrangements with, or they may contribute to and be rewarded by a common pool or pools.
    iv. Distribution preferences may be expressed by consumers in the name request. For example: SPAN://warnerbros/batman/directorscut/4K/<myaddress>/telstra/akamai/ (actual names and order may vary due to naming and routing conventions)
    v. Publishers may choose default delivery partners. In the event of a conflict, SPAN-AI and AI-HARD will choose the most efficient path, once again incentivising efficiency.
    vi. If the consumer or publisher does not specify distribution preferences, SPAN-AI and AI-HARD choose the most efficient path.
    vii. If a consumer's or publisher's chosen distributor or distributors is/are not the most efficient in any routing case, SPAN-AI and AI-HARD will choose the most efficient path and inform all interested parties of the decision to allow them to optimise efficiency.
    viii. Payment for distribution is calculated and made by a settlement system or systems, informed by the SPAN-AI and AI-HARD routing system, in a similar manner to how telephony call settlement is performed today.
    ix. Anyone may contribute resources and be rewarded for that contribution, providing a free market for telecomm's services. SPAN-AI monitors and maintains the security and health of the network. Non-performing resources will be removed. SPAN-AI is designed to work and meet QoS levels on both commercial and telco grade resources. QoS metrics and cost will determine the resources used and vice versa.

SPAN-AI uses an AI-driven Pub/Sub system for asynchronous multi-party communication services that support control plane dissemination: directory updates (names, discovery, configuration) and intelligence updates (optimization/control operations); as well as data plane dissemination: collaborative applications (messaging, video-conferencing, social networks, blogs, forums).

4 Scalable Pub/Sub System
   a. Fast, scalable, asynchronous, multi-party dissemination services
   b. Pubsub for disseminating control messages
      i. Name directory updates
      ii. Discovery updates (new peers, new services)
      iii. Configuration updates (new roles, new memberships)
      iv. Intelligence updates (optimization/control commands, e.g., resource allocation, storage, and routing decisions)
   c. Pubsub for disseminating application data
      i. Collaborative media apps
      ii. Social networks, blogs, forums
   d. The pub/sub system uses a new pub/sub algorithm, evolved from existing solutions (e.g., Gossipsub, HyParView, Plumtree, GoCast), that leverages the **AmI-Rendezvous service for improved operation**
      i. AmI-Rendezvous smart discovery and self-healing improves scalability and churn-resilience with little impact on routing scheme, other than tuning overlay degree, fanout, and probability weights.
      ii. Embedded plugins for self-healing and smart discovery strengthen peer discovery, activation, and lifecycle management of overlay:
         1. Pubsub protocol is embedded with pluggable metric aggregators, actuators, and triage for smart discovery
         2. A periodic heartbeat disseminates mesh-health metrics to AmI-Rendezvous
         3. Aggregated metrics are used by ML models to classify, rank, and recommend peer connections

SPAN-AI incorporates security integrated at all levels. SPAN-AI uses machine learning and recognition to detect and manage security threats. Content can be encrypted using commercial DRM systems such as PlayReady before it is published to the system. Data packets can be cryptographically signed by the publisher. Naming is rooted in self-sovereign identity, which can be defined as a lifetime portable digital identity that does not depend on any centralized authority. It uses decentralized identifiers that provide: persistence, global resolvability, cryptographic verifiability, and decentralization. Names can also be self certifying. Security is based on a hardware root of trust and secure boot and Web of Trust methods. Quantum encryption, i.e. encryption based on quantum state random number generators, may also be used.

5 Overarching security architecture and preferred embodiments
   a. Machine learning and recognition to detect and manage security threats
   b. Cryptographic packet signing
   c. Encryption and DRM
   d. Hardware root of trust
   e. Secure boot
   f. Decentralised identifiers. This may make use of Web of Trust methods.
   g. Sovereign identity
   h. Naming rooted in sovereign identity

i. Self-certifying names using CIDs with a prefix.
j. Quantum encryption

SPAN-AI Intelligence

SPAN-AI orchestrates the adaptive operation of the routing and pub/sub systems via a family of pluggable, hierarchical (local/edge/global) AI agents that provide monitoring, prediction, optimization, and control services with varying degrees of awareness and optimization capabilities at peer, edge, and core network levels.

SPAN-AI leverages a simulation, training, and development pipeline that enables cloud-level replication of runtime environments, simulation, testing, and training of AI models, that can then be plugged into peer/edge/core network nodes for real-time optimization and control.

The long-term goal of SPAN-AI is to provide a marketplace for plugable AI agents to enable open, flexible innovation in the optimization and control of universal networks. This may be based on a crypto token such as Filecoin or Blust.

1. Hierarchical AI
    a. Hybrid local/global optimization and control
        i. Combine fast local reactive self-organization with slower global/hierarchical proactive guidance/supervision and backup support
    b. Hierarchical intelligence
        i. Local intelligence at peer level
            1. Local monitoring and fast reactions for basic survivable operation and self-organization
            2. Fast and simple rules (e.g., filters, thresholds)
            3. Limited capability nodes
        ii. Ambient/swarm intelligence at edge level
            1. Higher level services supporting discovery, bootstrap, configuration, resource allocation, role assignment, storage decisions, routing hints, pub/sub memberships, naming
            2. Ideal place for ML models
            3. Nodes with higher capabilities, trust, and stability
        iii. Global intelligence at core level
            1. Optimization with global view
            2. Training of ML models to be pushed down to the edge
            3. AI Simulation
            4. HIghest capability nodes (e.g., stable peers, cloud nodes, ISP core, CDN PoP)
    c. Application and network awareness
        i. Global predictive knowledge of users' consumption/production patterns, application requirements, network conditions (including overlay mesh health), and available resources for proactive optimization

ii. Complemented by local situational awareness (of peer/mesh/network conditions) for reactive control and resilience to unpredictable change
iii. Exploit metadata in content requests (e.g., delivery deadlines)

d. Objectives & Principles
   i. No single point of failure. Fully decentralized operation with hierarchical fallback support
   ii. Metrics/actuator rules based ontology for data-driven innovation in reinforcement learning
   iii. Open framework with pluggable programming APIs
   iv. AI agent reuse and market enablement. Enable reuse of AI agents/components by innovative developers.
   v. Inter-operable with existing telco, ISP, CDN, and Internet networks. This may use existing standards such as OpenFlow or new AI interoperability standards evolved from the SPAN-AI ontologies
   vi. The ultimate objective is an inter-operable mesh of intelligent networks, operating as a single Universal Content Distribution Network, with no single point of failure

2. AI-HARD operation (intelligent hybrid adaptive routing)
   a. AI-HARD allows adaptively operating networks as p2p overlays, network-level meshes, or anything in between, via integration of NRR and NBR subsystems.
   b. Optimized role assignment and resource allocation
      i. Heterogeneous agents/roles
         1. Algorithms determine
            a. Service roles: DHT routing, name-based routing, storage, caching, discovery, monitoring, information mediation/decisioning
            b. Intelligence capability: reactive/proactive, local/global view, learning/observing, heuristic/optimization
         2. Based on
            a. Network architecture level: peers, gateways, servers, switches, routers, application servers, etc.
            b. Trust, security, and stability level
      ii. Resource allocation
         1. Algorithms determine
            a. Allocation of CPU, memory, disk, upload/download bandwidth resources to each subsystem within each agent
         2. Based on
            a. Service role
            b. Intelligence capability
            c. Network architecture level
            d. Trust, security, and stability level
         3. More stable, endpoint nodes will tend to have more resources allocated to storage-centric subsystem

4. Less stable, in-network nodes will tend to have more resources allocated to delivery-centric subsystem
5. Replication level in DHT is relaxed because name-based routing kicks in for delivery-centric applications
6. Name-based routing forwarding state is relaxed because DHT handles content with looser delivery requirements

iii. Incentivize resource contribution via FIL-type marketplace (see market enablement section)

iv. Mostly provided by ambient and global intelligence nodes with edge/core level view/capabilities

v. Complemented by self role assignment and resource allocation capabilities

c. Optimized long-term placement and short-term caching

i. Distributed cloud-network flow algorithms for long-term placement in distributed storage-centric subsystem

ii. Probabilistic local caching policies for delivery-centric subsystem

d. Adaptive name resolution vs name based routing

i. AI-based optimization and self-organizing methods determine how applications split requests into the two subsystems

1. Requests with loose delivery deadlines and predicted or pre-established requests with tight delivery deadlines can be handled by “slower”, but persistent and scalable, DHT subsystem
2. Requests with tight delivery deadlines are handled via fast name-based routing subsystem
3. Incentivize pre-planned requests via FIL-like marketplace (e.g., for big releases - see market enablement ection)

ii. DHT routing operation

1. Kademlia-type protocol but on improved multi-level DHT
2. Adaptive DHT parameter optimization (nodes, blocks, replication factor, concurrency factor)
3. AmI-Rendezvous for DHT maintenance and self-organization under high churn (see AmI-Rendezvous operation)

iii. Name based routing operation

1. NDN-type methods but with reduced forwarding state due to DHT subsystem cooperation

e. Inter-operable with existing telco, ISP, CDN, and Internet networks using existing IP protocols

f. Supporting IP routing alongside and compatible with NRR and NBR routing

g. The ultimate objective is an inter-operable mesh of intelligent networks, operating as a single Unified Content Distribution Network, with no single point of failure

3. AmI-Rendezvous operation (intelligent discovery and configuration)

a. Ambient Intelligence (AmI) refers to a combination of awareness and control for:

i. Peer/Local Intelligence: Embedded actuators in peers control probability weights, degree, and fanout of meshes (pubsub, NRR, NBR). Observers of p2p messaging compile metrics from neighbour actions and events to infer health (e.g., hop-count, reliability, latency, load-balance). Self-healing strategies can be as simple as filtering.

    ii. Edge/Swarm Intelligence: AmI health classification decisions (scoring, ranking) for peers and p2p overlay meshes are derived by a basic reinforcement learning model.
    iii. Core/Global Intelligence: Maintain aggregated usage predictions and mesh/network conditions. Determine placement of rendezvous servers.

b. AmI-Rendezvous service builds on state-of-the-art rendezvous services such as libp2p Rendezvous, which supports periodic peer re-registration, discovery, bootstrap, expanded with peer heartbeat and mesh health metrics and rankings.
c. A pluggable interface for self-healing agents embeds AmI-Rendezvous clients into peers
    i. Embeds pluggable metrics, actuators, and triage for smart discovery
    ii. A periodic heartbeat disseminates mesh-health metrics and change deltas to AmI-Rendezvous and SPAN-AI data lake
    iii. Registration and re-registration is extended to exchange whole mesh snapshots
d. A pluggable interface integrates smart discovery at rendezvous points
    i. AmI-Rendezvous server and messaging builds on libp2p Rendezvous service, with metrics collection expanded with testlab Prometheus and Influx.
    ii. Integrates reinforcement learning agents for mesh health classification.
    iii. Discovery is extended with peer rankings.
e. Additional features include discovery records, federation and caching, adaptive control, metrics/actuator reuse, topic specific/device specific metrics.
f. Further embedded intelligence and adaptive control of simulations and development pipeline by integrating with AI-HARD solution to support:
    i. hybrid P2P routing via rendezvous bypass to satisfy deadlines or optimise overlay breadth,
    ii. rendezvous & information mediator role assignment/placement,
    iii. partitioning by DHT topological layers, hints from naming enhancements or global awareness,
    iv. coarse grained adaptive control activating AI plugins,
    v. Smart discovery/healing for NRR and NBR routing.
g. Support different mesh types (NBR, NRR) and assurance requirements metrics (security, trust, integrity, efficiency, reliability, stability, latency).

SPAN-AI Simulator

SPAN-AI leverages a simulation, training, and development pipeline that enables cloud-level replication of runtime environments, simulation, testing, and training of AI models, that can then be plugged into peer/edge/core network nodes for real-time optimization and control.

SPAN-AI's simulator, **Self-Aware Mesh Simulator (SAMSim),** is supported by:

- distributed cloud hosting a **big data lake of meshes with health metrics**
- simulating and deploying AI models across an **automated software engineering pipeline**

1. Simulator Intelligence
    a. Test harness leverages SPAN-AI intelligence for adaptive scalability, including guiding placement of rendezvous servers, scheduling and supporting data exchange between federated rendezvous servers and central data lake.
    b. SAMSim provides standardised metrics for benchmarking while leveraging agile infrastructure and container deployment concepts from current benchmarking pipelines with an aim to balance scalability and realism.
    c. AI developers use mesh-health metrics to test, train, sample simulations and reward reinforcement learning.
        a. Mesh health metrics and actuators frame a reinforcement learning problem.
        b. Metrics and actuators are refined with rules elicitation and management in data lake.
        c. Agents are refined from rule sets, metrics, and actuators, into:
            i. mock control and protocol plugins for abstract simulators (e.g. PeerSim, agent-based simulation, Topology Simulator, existing Gerbil simulator)
            ii. prototype plugins for concrete embedded simulations (borrowing from D-P2P-Sim, RealPeer, ProtoSim, Testground) that are distributed across cloud providers like Amazon Web Services (AWS) or Digital Ocean Cloud (DO-Cloud).
        d. Data engineering tools are integrated to prepare and quality control datasets.
    b. Simulator & pipeline features:
        a. Mostly derived from integrating test framework and integrated P2P simulators
        b. Include multithreading, execution controls/runners/schedulers, realistic stubs and network models, agile container infrastructure, metrics, and visualisation.
    c. Additional simulators include NS3 for network events, RealPeer and ProtoSim for refinement model, PlanetSim for intelligent swarming, PEERFACTSIM.kom for DHT routing.
    d. Additional simulator & pipeline features include customisable API standards, simulator layering, topology graph export formats and failure simulation.

2. Pub/Sub Simulation
    a. Supports essential scaffolding to engineer various trial agents by evaluating and integrating capabilities from currently available p2p simulators (e.g. PeerSim, D-P2P-Sim).
    b. The agent engineering and assurance environment:
        i. Initially focuses on ensuring latency in scalable, resilient pubsub meshes.

ii. Supports simulation and iterative development of
    1. health metric compilation, triage for smart discovery on broadcast and gossip, triggering of actuators to filter or limit or weight probability of mesh overlays.
    2. messaging for AmIRendezvous registration, heartbeat, smart discovery
    3. messaging for pubsub bootstrap, subscription/publication, grafts/purges

iii. Supports reuse of mesh health data sets, health metrics and healing actuator rules across empirical simulation experiments. Reuse may be extended in future embodiments.

SPAN-AI use cases and development roadmap:

1. Release 1
    a. MVP to meet requirements of Protocol Labs RFPs 7 and 8
2. Release 2
    a. Additional R&D as specified in SPAN-AI architecture
3. Release 3
    a. Physical PoC implementing SPAN-AI-HARD on switches, edge servers and peers + additional R&D
    b. Peers may be physical or agents deployed on cloud services to simulate peers
4. Release 4
    a. Baseline production MVP + additional R&D
5. Release TBA
    a. Distributed video origin store and delivery service
        i. Ingest video: encode; package; encrypt; sign
        ii. Assign unified names
        iii. Publish to distributed storage network (NRR) with storage metrics
        iv. Publish in universal pub/sub system with distribution QoS metrics
    b. Subscribe to a video using universal pub/sub system
    c. Distribute video using NBR network for real time (live) streaming and/or NRR network for near/non real time distribution e.g. on-demand

-END-

GLOSSARY

Peer: any hardware or software apparatus with a similar or comparable general or specific purpose in whole or part.

P2P: Peer-to-Peer.

Agent: a software application with varying degrees of awareness, communication, optimization, learning, reporting, self-organising or other capabilities distributed to and/or running on any network appliance (computer, consumer electronics device, router, switch, server, etc) at peer, edge, core, or other network levels; a virtual network service or application. This could be an AI agent; an application running on a virtual peer e.g. an operating system running in a virtual environment; a network service; etc.

I/F: interface. A method of inter-connecting software or hardware applications to each other or to people for the purpose of communication. In a preferred embodiment the method is open and standardised in which case the interface may be known as an Application Programming Interface or API.

Peer Network: any network with a similar or comparable general or specific purpose in whole or part.

IP: Internet Protocol; the “thin waist” routing protocol of the original and current Internet

TCP: Transport Control Protocol

SPAN: Secure Peer-Assisted Networking

AI: Artificial Intelligence

ML: Machine Learning

AmI: Ambient Intelligence

HARD: Hybrid Adaptive Routing Design

SAMSim: Self-Aware Mesh Simulator

CID: Content Identifier

IPFS: Inter-Planetary File System

IPLD: Inter-Planetary Linked Data

IPNS: Inter-Planetary Name System

DNS: Domain Name System

DNSLink: protocol that uses DNS text records to link domain names to IPFS addresses or CIDs

NDNS: Domain Name System for Named Data Networking

mDNS: multicast DNS

Pub/Sub: Publish/Subscribe

libp2p: a location independent modular network stack. Part of IPFS.

NRR: Name Resolution based Routing

NBR: Name Based Routing

NDN: Named Data Networking

NBN: either Name Based Networking or National Broadband Network in Australia

DHT: Distributed Hash Table

DRM: Digital Rights Management

VoD: Video on Demand

ISP: Internet Service Provider

CDN: Content Distribution Network

PoP: Point of Presence

FIL: Filecoin crypto token trading abbreviation

testlab & testground: IPFS test frameworks

PoC: Proof of Concept

MVP: Minimum Viable Product

NRT: Near Real Time or Non-Real Time

ISO: International Standards Organisation

QoS: Quality of Service

telecom's: telecommunications

telco: telecommunications company

Node: a vertex of a graph network model; the joining point of graph edges;

Edge: network edge (1-2 hops away from the end-user device); or the connection between nodes in a graph;

Graph: mathematical model used to represent communication networks, data organization, computational devices, the flow of computation or communication, etc.

UND: Unified Naming and Discovery/Directory system/service

