[SIG-NETWORK]

Gateway API: Multi Port Approach

Authors: Rob Scott , <You>

Background

Currently we require a single port to be specified with each Gateway listener. This prevents a number of use cases, including:

1. Listen on all ports and protocols
2. Listen on broad range of ports
3. Auto assign port when none specified

This doc intends to show how we could support these use cases within Gateway API. As we consider graduating a Gateway resource to beta, we want to understand what options we'd have to add support for this in the future without a breaking change.

For additional context, these concepts have also been explored in upstream Kubernetes. Network Policy has recently added support for port ranges (docs, KEP). Similarly, some initial work has gone into exploring what supporting all ports could look like for Services (issue, KEP).

Compatible Approach: Make Port Optional, add new EndPort Field

We can support all of the use cases described above by copying what NetworkPolicy did, enabling consistency across the APIs. The following states would be valid:

A. *Port is nil*
   Support: Extended. Where supported, a port will be automatically assigned.

B. *Port is set, EndPort is nil*
   Support: Core. Functionality will match existing behavior. The listener will listen on the single

C. *Port and EndPort are set*
   Support: Extended. This represents a port range. When supported, a listener will listen on all ports within this range. This can be used to represent the "All Ports" use case by using min and max port numbers.

Validation:

- EndPort can only be defined if Port is also defined
- EndPort must be greater than Port

Status:

If we added support for automatic port assignment, a new AssignedPort(s) field would need to be added to ListenerStatus. New conditions would be defined when a configuration is present

that is not supported by an implementation, specifically automatic port assignment or port ranges.

Compatibility:
This change is considered backwards compatible. It makes a required field optional and adds a new optional field.

Timeline:
Since this can be a backwards compatible change, this does not need to block beta and can be added at a later point. Each individual part of this can be tackled independently depending on if/when we are able to build consensus for a feature. (Making port optional can be done separately from adding a new EndPort field).

Alternatives Considered

1. Make Port Optional
Solves use case 1 or 3, unclear if we'd want to leave interpretation of empty/unspecified port this open ended though.

2. Make Ports a List
Solves use case 2, does not help with 1 or 3.