

POLICY SERVICE SYSTEM ARCHITECTURE FOR SESSIONS CREATED USING STUN

BACKGROUND

[0001] As communication technology continues to expand, it may be desirable to bring
5 together many different forms of communication such as Short Message Services (SMS),
Multimedia Message Services (MMS), electronic messaging (email), instant messaging
(IM), online presence status, as well as voice and video communications over a network.
One such communication type gaining popularity is voice over Internet protocol (VoIP)
sessions. Traditionally, VoIP systems interact with traditional public switched telephone
10 networks (PSTN) to enable users to place telephone calls using traditional landline
telephones, mobile phones, and other communication devices. Still other communication
devices, such as microphones and cameras attached to various computing devices, such as
personal computers, also enable VoIP sessions to be established.

[0002] One drawback of VoIP is that when voice and/or video data is transmitted over a
15 packet switched network, it competes with other network resources that are being used by
traditional data applications. If the network is overloaded there may not be enough
bandwidth for each application. As a result, not only will data access and transfers by the
traditional applications be affected, but the quality of service of the VoIP session may also
be diminished.

[0003] It is with respect to these and other considerations that embodiments of the
20 present invention have been made. Also, although relatively specific problems have been
discussed, it should be understood that embodiments of the present invention should not
be limited to solving the specific problems identified in the background.

SUMMARY

[0004] This summary is provided to introduce a selection of concepts in a simplified
25 form that are further described below in the Detail Description section. This summary is
not intended to identify key features or essential features of the claimed subject matter, nor
is it intended to be used as an aid in determining the scope of the claimed subject matter.

[0005] Described herein are embodiments for setting, managing, and monitoring
30 connection policies for a plurality of network paths from a first endpoint to a second
endpoint. In embodiments, a set of candidate addresses from a first endpoint and a set of
candidate addresses from a second endpoint are received. Each candidate address of the
first endpoint is paired with each candidate address of the second endpoint to create a

plurality of candidate address pairs. Each candidate address pair has an associated network path. The network path is the route by which data flows to and from the various endpoints. Once all network paths between each candidate address pair have been determined, a connection policy is set for each network address pair and the associated network path.

[0006] Embodiments may be implemented as a computer process, a computing system or as an article of manufacture such as a computer program product or computer readable media. The computer program product may be a computer storage media readable by a computer system and encoding a computer program of instructions for executing a computer process. The computer program product may also be a propagated signal on a carrier readable by a computing system and encoding a computer program of instructions for executing a computer process.

BRIEF DESCRIPTION OF THE DRAWINGS

[0007] Embodiments of the present disclosure may be more readily described by reference to the accompanying drawings in which like numbers refer to like items and in which:

[0008] FIG. 1 illustrates a block diagram of a system for setting connection policies for a plurality of network paths according to an embodiment.

[0009] FIG. 2 is an operational flow for setting connection policies for a plurality of network paths according to an embodiment.

[0010] FIGS. 3A and 3B illustrate sample candidate address lists and the resulting candidate address pairs according to an embodiment.

[0011] FIG. 4 illustrates sample network paths between a first endpoint and a second endpoint according to an embodiment.

[0012] FIG. 5 illustrates a computing environment for implementing embodiments disclosed herein.

DETAILED DESCRIPTION

[0013] Various embodiments are described more fully below with reference to the accompanying drawings, which form a part hereof, and which show specific embodiments for practicing the invention. However, embodiments may be implemented in many different forms and should not be construed as limited to the embodiments set forth herein; rather, these embodiments are provided so that this disclosure will be thorough and complete, and will fully convey the scope of the invention to those skilled in the art. Embodiments may be practiced as methods, systems or devices. Accordingly,

embodiments may take the form of a hardware implementation, an entirely software implementation or an implementation combining software and hardware aspects.

[0014] Embodiments generally relate to setting, managing and monitoring connection policies for network paths between two or more endpoints. According to embodiments,

5 each endpoint has one or more candidate addresses. Each candidate address from each of the different endpoints are paired together to create candidate address pairs. A network path that connects the endpoints via the candidate address pairs is then determined and each network path is assigned a policy. The network paths may be used to transfer data between the two endpoints. In embodiments, the data transfer between the endpoints is
10 governed by the policy. For example, a policy may dictate what type of data may be transferred on a network path or the policy may dictate how much bandwidth may be used for the data transfer.

[0015] FIG. 1 illustrates a system for setting, maintaining and monitoring connection policies for various network paths that exist between two endpoints according to

15 embodiments. As shown in FIG. 1 the system may contain a sending endpoint 110 and a receiving endpoint 130. In an embodiment, the sending endpoint 110 attempts to establish a network session with a receiving endpoint 130. The sending endpoint 110 and the receiving endpoint 130 may be one of a telephone, a mobile or handheld device, a personal computer or laptop, or any other computing device capable transmitting and/or receiving
20 data over a network (not shown). The session may be a voice over Internet Protocol (VoIP) telephony session. As briefly discussed above, VoIP enables voice communications, as well as other data, such as video, to be transmitted over the internet or other networks to a various endpoints. The endpoints may be local endpoints or remote endpoints. Although a VoIP telephony session is specifically mentioned herein, the
25 present specification is not limited to such sessions. It is contemplated that any type of data transfer between endpoints may utilize the methods of the present disclosure.

[0016] The sending endpoint 110 may have one or more candidate addresses (e.g., Internet Protocol (IP) addresses) that it may use to establish a data transfer session with a second endpoint. Each endpoint may have the multiple candidate addresses by having
30 more than one local IP address available, being behind a Network Address Translator (NAT), or using the services of a STUN server. As is known in the art, an IP address is a numerical identifier assigned to a device that participates in network session. The IP address uniquely identifies the particular device to/from which data is being sent and received.

[0017] In embodiments, when attempting to establish a session with a remote endpoint, such as the receiving endpoint 130, the sending endpoint 110 transmits a request 111, or signaling session, through a signaling layer of the network. The request includes data for setting up and negotiating a session between the sending endpoint 110 and the receiving
5 endpoint 130. The request 111 may be routed through a server 120. According to an embodiment, the server 120 may include a signaling server configured to route the request between the sending endpoint 110 and the receiving endpoint 130. Server 120 may also include a relay server configured to provide additional candidate address to either endpoint that may be used for exchanging data within the session. In another embodiment the relay
10 server and signaling server may be separate entities. The server 120 forwards the request 111 to the receiving endpoint 130. The resulting signaling negotiation session between the two endpoints may rely on a signaling layer protocol session such as the Session Initiation Protocol (SIP).

[0018] The information sent by the sending endpoint 110 also includes, with the request
15 111, a list of candidate addresses 112. The candidate addresses 112 are IP addresses associated with the sending endpoint 110. In an embodiment, the sending endpoint 110 performs a candidate discovery phase during which the sending endpoint 110 discovers IP addresses it may use to establish a network session with another endpoint. Each candidate address may be used to establish network sessions with, and transmit data to/from, local
20 and remote endpoints.

[0019] As briefly discussed, the system may also include a server 120. According to embodiments, the server 120 may include a signaling server configured to forward the request 111, along with the candidate addresses 112 of the sending endpoint 110, to the receiving endpoint 130. In embodiments, the receiving endpoint 130 may be an endpoint
25 in the same private network as the sending endpoint 130. Alternatively, the receiving endpoint may be outside the network of the sending endpoint 110. Thus, the request 111 may need to make multiple hops along a network path before reaching the receiving endpoint 130. In an embodiment, the server 120 may also include a relay server configured to provide additional candidate addresses to either the sending endpoint 110 or
30 the receiving endpoint 130. Depending on which endpoint receives the additional addresses, the additional addresses are appended to either endpoint's list of candidate addresses.

[0020] Once the session request 111 and the candidate addresses 112 of the sending endpoint 110 have been received by the receiving endpoint 130, the receiving endpoint

130 begins a candidate discovery phase. In the candidate discovery phase, the receiving endpoint 130 determines its own local candidate addresses 132. As part of the candidate discovery phase, the receiving endpoint 130 may discover additional candidate addresses it can use. These additional candidate addresses may be provided by the server 120 or by another server in the system.

[0021] Upon completion of the candidate discovery phase, the receiving endpoint 130 transmits a request, the list of candidate addresses 112 from the sending endpoint 110, and its own candidate addresses 132 to a server 140. The request may be the session request 111 first transmitted by the sending endpoint 110. The request may also include a request to allocate the candidate addresses 132 of the receiving endpoint 130 on the server 140.

[0022] According to embodiments, server 140 may be a STUN (Session Traversal Utilities for NAT (Network Address Translator)) server. In embodiments, the STUN server is configured to receive both sets of candidate addresses, (e.g., candidate addresses 112 from the sending endpoint as well as the candidate addresses 132 of the receiving endpoint 130) and generate a set of candidate address pairs 142. The candidate address pairs 142 are generated by combining one of the candidate addresses 112 from the sending endpoint 110 with one of the local candidate addresses 132 of the receiving endpoint 130. In embodiments, each candidate address of the sending endpoint 110 is paired with each local candidate address of the receiving endpoint 130. This will be further explained below with reference to FIGS. 3A and 3B.

[0023] In embodiments, the STUN server 140 may also have a pool of its own local addresses. When requested, the STUN server 140 will allocate some, or all, of its STUN server addresses to the requesting endpoint 130. In response to the request, the STUN server addresses are added to the list of local candidate addresses 132 of the sending endpoint 130. Therefore, when a session is established between the sending endpoint 110 and the receiving endpoint 130, the addresses known by the STUN server 140 may also be used as candidate addresses of the receiving endpoint 130. In yet other embodiments, the STUN server 140 may also add any NAT addresses to the list of candidate addresses 132 of receiving endpoint 130. When the request 111 transmitted from the receiving endpoint 130 to the STUN server 140 passes through any NATs, each NAT address will also be considered as part of the receiving endpoint's 130 candidate addresses 132.

[0024] In other embodiments, receiving endpoint 130 may be required to transmit session data 134 to the STUN server 140. Session data 134 contains information regarding the requested session type between the sending endpoint 110 and the receiving

endpoint 130. Many different session types may be requested between the endpoints. Non-limiting examples include a VoIP session, a video session, or a combination VoIP/video session.

[0025] Session data 134 may also include a request for bandwidth. Depending on the type of session requested (e.g., VoIP, video, etc.) either the sending endpoint 110 or the receiving endpoint 130 may request that a certain amount of bandwidth be allocated on a network path for the requested session. For example, if the requested session type is a VoIP session, the session data 134 may include a bandwidth request for 50 kbps. If however, the requested session type is a VoIP session with corresponding video data, the requested bandwidth may be 100 kbps.

[0026] Upon completion of the candidate address pair 142 generation, the list of candidate address pairs 142 is transmitted to a policy server 150. Based on configuration information corresponding to each candidate address pair, the policy server 150 may determine the subnet that each candidate address of each candidate address pair belongs to. The policy server 150 may map each subnet to various locations and determine the network path that connects the endpoints represented by the candidate address pair.

[0027] In an alternative embodiment, the receiving endpoint 130 may contact the STUN server 140 to discover the server addresses of the STUN server 140 as well as any NAT addresses known by the STUN server 140. When the receiving endpoint 130 has received these additional addresses, the receiving endpoint 130 may send these addresses directly to the policy server 150, bypassing the candidate addresses generation phase. The policy server 150 may then determine the network paths that connect each endpoint and determine a policy for each path as described below.

[0028] In embodiments, the policy server 150 sets one or more policies for each network path associated with each of the candidate address pairs 142. According to embodiments, the policies are network use policies. For example, the policies may indicate a total amount of bandwidth allowed on each network path, a maximum amount of bandwidth allocated for each session using the network path, a total number of data transfers or phone calls permitted on the particular network path, or any combination thereof. Embodiments provide that each policy is configured, monitored, and maintained by an IT administrator.

[0029] In embodiments, the session data 134 may also be transmitted from the STUN server 140 to the policy server 150. As explained above, the session data 134 may include the session type, a requested amount of bandwidth, or a combination thereof. Policy server 150 may be configured to analyze the requested session type and determine which

candidate address pairs and associated policies will allow the requested session type. The policy server 150 also determines whether each network path associated with the candidate address pairs allows requested bandwidth amount to be allocated. In yet another embodiment, a policy may prohibit the use of certain candidate address pairs. In such cases, these candidate address pairs pre-fail any connectivity checks that will be performed by the receiving endpoint 130 as will be described in greater detail below.

[0030] When these determinations have been made, the policy server 150 transmits policy data 152 to the STUN server 140. The STUN server 140 then transmits, in a response message, the policy data 152 to the receiving endpoint 130.

[0031] In embodiments, policy data 152 includes the candidate address pairs 142, the associated network paths, and the network use policies for each path. Because each endpoint potentially has multiple candidate addresses both the sending and receiving endpoints may have multiple network paths to choose from when establishing the session. Having multiple network paths available provides a fall back mechanism in situations in which one of the network paths fail, when a policy of one of the network paths does not permit the requested connection type, or when one of the network paths does not have the requested resources available. For example, if a requested amount of bandwidth is not available on one network path, another network path with the available bandwidth may be selected.

[0032] In another embodiment, policy data 152 may only include candidate address pairs having policies that allow the requested session type indicated by the policy data 152. For example, if the requested session type included a session for transferring both audio and video data, the policy server 150 would only include the candidate address pairs having policies that allow the transfer of both audio and video data across a network path.

[0033] Other embodiments provide that policy server 150 may generate an authorization token 154. The authorization token 154 is transmitted, along with the policy data 152, to the server 120 (e.g. signaling server). The authorization token ensures that endpoints cannot subvert policy and continue to setup a session without a valid token. Acceptance of the authorization token 154 by either endpoint may be required to complete the negotiation session between the endpoints. In embodiments, the authorization token 154 is transmitted to the sending endpoint 110 from the receiving endpoint 130 as part of a response to the signaling request 111.

[0034] When receiving endpoint 130 receives a response, including the policy data 152 from the STUN server 140, the receiving endpoint 130 may perform a STUN connectivity

check. As part of the connectivity check, any network paths that have policies that do not allow the requested session will automatically fail. Additionally, if policy prohibits the use of a particular candidate pair, that particular candidate address pair will have already pre-failed the connectivity check. Receiving endpoint 130 will transmit only allowable
5 addresses 136 (e.g., the network address pairs and the associated network paths that allow the requested session type) to sending endpoint 110. Thus, a session may be negotiated between the two endpoints using only network paths that support the requested session type.

[0035] In yet other embodiments, the receiving endpoint 130 and the sending endpoint
10 110 may each perform connectivity checks. Because each endpoint may perform connectivity checks, cases may arise where one endpoint (e.g. the sending endpoint 110) does not know that a particular candidate address pair has pre-failed a connectivity check performed at the other endpoint (e.g., the receiving endpoint 130). In such cases, the sending endpoint 110 may still attempt a connectivity check using the pre-failed candidate
15 address pair. When the connectivity check is performed on the pre-failed candidate address pair at the sending endpoint 110, the receiving endpoint 130 will ignore the attempted connectivity check associated with the pre-failed candidate address pair. Because the connectivity check was ignored at the receiving end, the sending endpoint 110 will treat the pre-failed candidate address pair as having failed the connectivity check.

20 Thus, the pre-failed candidate address pair will not be used for the session.

[0036] Embodiments provide that either the sending endpoint 110 or the receiving endpoint 130 may select which network path of the multiple network paths is to be used for the session. The determination may be made based on the first identified network path that permits the requested session. Alternatively, a ratings system may be implemented in
25 which the network paths are given a quality of service rating by the policy server. The rating factors may include session quality metrics monitored by a system module. In yet other embodiments, the rating may be based on whether each network path allows the requested session type and/or has the requested bandwidth available. In still yet other embodiments, the ratings system may be a combination of any of the above.

30 **[0037]** It is also contemplated that some requested session types may not be allowed on a network path due to policy restrictions or the requested bandwidth is simply not available on any of the network paths. For example, session data 134 may indicate that an audio/video session is the requested session type. Furthermore, the session data 134 may request that a transfer rate of 100 kbps be allocated on the network path for the

audio/video session. Once all address pairs have been evaluated and their respective policies determined, it may be discovered that none of the resulting candidate address pairs, nor their associated network paths, allow either the requested session type (e.g. audio/video session) or have the requested bandwidth (e.g., 100 kbps) available.

5 **[0038]** In such instances, embodiments provide that either the receiving endpoint 130 or the sending endpoint 110 may modify the requested session type and/or the requested bandwidth to bring the request within the parameters of at least one of the policies associated with one of the network paths. Continuing the example from above, the receiving endpoint 130 may modify the session type to be an audio session only. As a
10 result, the receiving endpoint 130 transmits to the sending endpoint 110 the allowable addresses 136 (e.g. candidate address pairs) having policies that allow for the requested audio session to occur.

[0039] Other embodiments provide that the system of FIG. 1 also includes a monitoring module 160. The monitoring module 160 is configured to monitor each network path
15 between the endpoints and report session quality metrics to the policy server 150. The monitoring module 160 may be configured to receive reports from the endpoints involved in the session as to the quality of the experience. Other embodiments provide that the monitoring module may also receive reports regarding whether the policy for the network path is being adhered to, the quality of service (QoS) on the network path, whether the
20 session is still established between the endpoints, an amount of bandwidth being used for each session using the network path, available bandwidth on the network path, and a total number of sessions currently using the network path. The module 160 enables the policy server 150 to make informed decisions when setting policies for new sessions that are to be established between a new set of endpoints having candidate addresses that will use or
25 cross the same network paths.

[0040] FIG. 2 illustrates an operational flow 200 for setting connection policies for a plurality of network paths according to an embodiment. Operational flow 200 may be performed in any suitable computing environment. For example, the operational flow may be executed by systems such as the system described in FIG. 1. Therefore, the description
30 of operational flow 200 may refer to at least one of the components of FIG. 1. However, any such reference to components of FIG. 1 are for descriptive purposes only.

[0041] Operational flow 200 begins at step 210 in which a receiving endpoint 130 (FIG. 1) receives a signaling request from a sending endpoint 110. Included with the signaling request are multiple candidate addresses available to the sending endpoint 110. According

to embodiments, the signaling request may be transmitted from the sending endpoint 110 to the receiving endpoint 130 via a signaling layer protocol such as the Session Initiation Protocol (SIP).

[0042] When the signaling request and candidate addresses have been received, step 220 provides that the receiving endpoint 130 enters into a candidate discovery phase. During the candidate discovery phase, the receiving endpoint 130 identifies all local candidate addresses known by the receiving endpoint 130.

[0043] Step 230 provides that the candidate addresses received from the sending endpoint 110 and the candidate addresses from the receiving endpoint 130 are received at the server. According to embodiments, the server is a STUN server 140. The sending endpoint 130 may, as part of the candidate discovery phase, transmit a request to the STUN server 140 requesting that the STUN server 140 allocate IP addresses stored on the STUN server as candidate addresses of the receiving endpoint 130. Thus, the receiving endpoint 130 can use both its own local candidate addresses as well as the IP addresses stored on the STUN server 140 when establishing a session. Additionally, step 230 provides that in the event the request transmitted from the receiving endpoint 130 to the STUN server 140 passed through a NAT, the STUN server also includes the NAT address as candidate addresses of the sending endpoint 130. As a result, the final list of the candidate addresses of the receiving endpoint 130 may contain the receiving endpoint's local candidate addresses, the STUN server's locally stored IP addresses, as well as any NAT addresses passed through by the request.

[0044] Step 230 also provides for generating a list of candidate address pairs.

According to embodiments, each candidate address in a list of candidate addresses from the sending endpoint 110 is paired with each candidate address from a list of candidate addresses of the receiving endpoint 130. As explained above, candidate addresses from the sending endpoint 110 may also be paired with the addresses allocated from the STUN server, addresses from a relay server, and/or any possible NAT addresses that have been included as candidate addresses of the receiving endpoint 130. The resulting candidate address pairs are an $n \times n$ matrix of candidate addresses, where each candidate address from one endpoint is paired with every other unique address from the second endpoint.

[0045] In embodiments, the STUN server may require that session data corresponding to the requested session type be received from one of the endpoints. The session data may also include a requested amount of bandwidth for the session, although it may not be required.

[0046] In embodiments where session data is required, flow may pass to step 235. Step 235 determines whether session data has been received at the STUN server 140. If session data, indicating at least the requested session type is not received, the session request ends and a session will not be established between the endpoints. In other embodiments, if
5 session data, indicating at least the session type is received, flow may proceed to step 240. In embodiments when a request for bandwidth has not been included in the session data, the session can use a maximum amount of bandwidth allocated by a policy for that particular type of session.

[0047] For example, if session data indicates that an audio session is the requested
10 session type but a bandwidth request has been omitted, the requested session type will be established on a network path having a policy that allows the requested session, but the system will only allocate a default amount of bandwidth (e.g., 50 kbps) for the session. In yet other embodiments, the STUN server may require that both the session type and requested/required bandwidth be contained in the session data or the request will fail.

[0048] In step 240, the candidate address pairs are transmitted to a policy server 150.
15 Step 250 provides that a network path for each candidate address pair is determined by the policy server 150. The policy server then determines and/or sets a policy for each network path and its associated candidate address pair. According to embodiments, the policy may set a maximum amount of bandwidth allowed for the requested session type. In other
20 embodiments the policies may set a maximum number of sessions or a maximum number of calls on a particular network path. Still other policies may govern what session types may be established on the various network paths.

[0049] When the policies have been set for each network path, step 260 provides that the
25 candidate address pairs, the associated network paths, along with the associated policies are returned to the STUN server 140. In embodiments, the STUN server 140 may compare the requested session type, indicated by the session data, with each of the network address pairs and the associated policies. If some of the network paths do not support the requested session type, or some of the network paths do not have the required/requested bandwidth available, or the network paths are currently unavailable,
30 these particular network paths will fail the STUN server's connectivity check. As a result, these particular network paths will be removed from the candidate address pairs (e.g. policy data 152) prior to being returned to the receiving endpoint 130. In alternative embodiments, all network address pairs are returned to the receiving endpoint 130 and the

connectivity checks are performed by the receiving endpoint 130. The resulting allowable addresses 136 may then be transmitted to the sending endpoint 110.

[0050] When the candidate address pairs have been transmitted to the receiving endpoint 130 from the STUN server 140, the receiving endpoint 130 may send each allowable address 136 to the sending endpoint 110. The allowable addresses 136 may include each network address pair and associated policy that permits the requested session type. The allowable addresses 136 may be included as a response to the signaling session that was first transmitted by the sending endpoint 110. Once received, the requested session is established using at least one of the network paths and its associated policy.

[0051] FIGS. 3A and 3B illustrate sample candidate address lists and the resulting candidate address pairs according to an embodiment. The sending endpoint may transmit sending endpoint candidate addresses 310 to a receiving endpoint. As shown in FIG 3A, the sending endpoint has three candidate addresses available to it when attempting to establish a session. Although the candidate addresses are depicted as IPv4 addresses, it is contemplated that IPv6 addresses may also be used. Additionally, it is contemplated that some of the candidate addresses may be IP addresses on a private network and are routed through a router or a hub. Session requests may be sent from one endpoint to another endpoint on the same private network.

[0052] Continuing with the above example, receiving endpoint candidate addresses 320 may also include three different candidate addresses. As with the sending endpoint candidate addresses 310, the receiving endpoint candidate addresses may also be IPv4 and/or IPv6 addresses.

[0053] According to embodiments, both sending endpoint candidate addresses 310 and receiving endpoint candidate addresses 320 are transmitted to a server, such as STUN server 140 (FIG. 1). When the candidate addresses are received by the STUN server 140, the STUN server 140 generates candidate address pairs 330. As shown, each candidate address from the sending endpoint candidate addresses 310 is paired with each candidate address from the receiving endpoint candidate 320. The resulting candidate address pairs 320 comprise a one to one mapping of all candidate addresses from both the sending and receiving endpoints.

[0054] FIG. 3B illustrates a scenario in which the receiving endpoint candidate addresses 320 has been expanded to include a candidate address (127.133.0.5) that was stored locally on a server, such as STUN server 140 (FIG. 1). Receiving endpoint candidate addresses 320 may also include any NAT addresses that a request may have

crossed as it was sent from the receiving endpoint to the STUN server. Much like the candidate address pairings explained above, each candidate address of the sending endpoint candidate addresses 310 is paired with each candidate address of the receiving endpoint candidate addresses 320 to create candidate address pairs 330. The candidate address pairs 330 may then be transmitted to a policy server as described above.

[0055] FIG. 4 illustrates sample network paths 400 and 420. According to embodiments, each network path 400 and 420 consists of one of the plurality of candidate address pairs 330 (FIGS. 3A-3B). For example, network path 400 comprises candidate address 208.77.188.166 from the sending endpoint and candidate address 172.184.5.10 from the receiving endpoint. As shown, network path may need to make any number of hops to traverse the network path from the sending endpoint to the receiving endpoint and vice versa.

[0056] In embodiments, each of the network paths 400 and 420 may have an associated policy (e.g. policy 410 and 430 respectively). The policy is set by the policy server.

When the policies are set for each network path, the candidate address pairs are returned to the receiving endpoint and a determination is made as to which network path will be used for the session. This determination may be made, based in part, on the policy associated with each of the network paths.

[0057] For example, session data may indicate that an audio/video type session is being requested. Session data may also indicate a request for bandwidth usage of 150 kbps.

When the candidate address pairs, network paths, and policies have been set as described above, the resulting candidate address pairs are returned to the endpoints. A determination is made as to which network path will be used for the requested session.

[0058] Continuing with the above example, the policy server may return network path 400 and its associated policy 410 to a receiving endpoint. The policy 410 may limit sessions on the network path 400 to audio sessions only. However, the policy may also indicate that sessions using this particular network path have been allocated bandwidth of 150 kbps. In contrast, the policy 430 for network path 420 may allow audio/video sessions but limit the bandwidth for the session to 100 kbps.

[0059] Because neither network path allows for the requested session type or the requested bandwidth, the session request may be tailored to fit within the bounds of either policy 410 or 430. For example, if quality of service is a main concern for the audio portion of the session and video data was not necessarily desired, the session request may be modified to negotiate only audio transfers from each endpoint. As a result, an audio

session may be established on network path 400. If however, video is required for the session, network path 420 may be selected and the session request may be modified to fall within the 100 kbps bandwidth requirement set by policy 430. In still yet other embodiments, the session request may only be required to specify the session type. In such embodiments, any network path that allows the specified type of session may be chosen for the session and the session request need not be modified.

[0060] When the session has been established between the sending endpoint and the receiving endpoint along a particular network path (e.g., network path 400), the network path may be monitored for bandwidth usage, quality of service and other session quality metrics. This data is transmitted to the policy server. The policy sever is enabled to make informed decisions about what policies to establish on the various network paths when new sessions requests are received and the policy server sets the new policies in light of any existing sessions using each network path.

[0061] With reference to FIG. 5, an embodiment of a computing environment for implementing the various embodiments described herein includes a computer system, such as computer system 500. Any and all components of the described embodiments may execute as or on a client computer system, a server computer system, a combination of client and server computer systems, sending and receiving endpoints, and other possible computing environments or systems described herein. As such, a basic computer system applicable to all these environments is described hereinafter.

[0062] In its most basic configuration, computer system 500 comprises at least one processing unit or processor 504 and system memory 506. The most basic configuration of the computer system 500 is illustrated in FIG. 5 by dashed line 502. In some embodiments, one or more components of the described system are loaded into system memory 506 and executed by the processing unit 504 from system memory 506.

Depending on the exact configuration and type of computer system 500, system memory 506 may be volatile (such as RAM), non-volatile (such as ROM, flash memory, etc.), or some combination of the two.

[0063] Additionally, computer system 500 may also have additional features/functionality. For example, computer system 500 includes additional storage media 508, such as removable and/or non-removable storage, including, but not limited to, magnetic or optical disks or tape. In some embodiments, software or executable code and any data used for the described system is permanently stored in storage media 508. Storage media 508 includes volatile and non-volatile, removable and non-removable

media implemented in any method or technology for storage of information such as computer readable instructions, data structures, program modules, or other data.

[0064] System memory 506 and storage media 508 are examples of computer storage media. Computer storage media includes RAM, ROM, EEPROM, flash memory or other memory technology, CD-ROM, digital versatile disks (“DVD”) or other optical storage, magnetic cassettes, magnetic tape, magnetic disk storage, other magnetic storage devices, or any other medium which is used to store the desired information and which is accessed by computer system 500 and processor 504. Any such computer storage media may be part of computer system 500. In embodiments, system memory 506 and/or storage media 508 stores data used to perform the methods or form the system(s) disclosed herein. In embodiments, system memory 506 stores information such as policy data 514 and/or candidate addresses 516.

[0065] Computer system 500 may also contain communications connection(s) 510 that allow the device to communicate with other devices. In embodiments, communications connection(s) 510 may be used to transmit and receive messages between sender devices, intermediary devices, and recipient devices. Communication connection(s) 510 is an example of communication media. Communication media may embody a modulated data signal, such as a carrier wave or other transport mechanism and includes any information delivery media, which may embody computer readable instructions, data structures, program modules, or other data in a modulated data signal. The term “modulated data signal” means a signal that has one or more of its characteristics set or changed in such a manner as to encode information or a message in the data signal. By way of example, and not limitation, communication media includes wired media such as a wired network or direct-wired connection, and wireless media such as an acoustic, RF, infrared, and other wireless media. In an embodiment, the methods described above may be transmitted over the communication connection(s) 510.

[0066] In some embodiments, computer system 500 also includes input and output connections 512, and interfaces and peripheral devices, such as a graphical user interface. Input device(s) are also referred to as user interface selection devices and include, but are not limited to, a keyboard, a mouse, a pen, a voice input device, a touch input device, etc. Output device(s) are also referred to as displays and include, but are not limited to, cathode ray tube displays, plasma screen displays, liquid crystal screen displays, speakers, printers, etc. These devices, either individually or in combination, connected to input and

output connections 512 are used to display the information as described herein. All these devices are well known in the art and need not be discussed at length here.

[0067] In some embodiments, the component described herein comprise such modules or instructions executable by computer system 500 that may be stored on computer storage medium and other tangible mediums and transmitted in communication media. Computer storage media includes volatile and non-volatile, removable and non-removable media implemented in any method or technology for storage of information such as computer readable instructions, data structures, program modules, or other data. Combinations of any of the above should also be included within the scope of readable media. In some embodiments, computer system 500 is part of a network that stores data in remote storage media for use by the computer system 500.

[0068] This disclosure described some embodiments of the present disclosure with reference to the accompanying drawings, in which only some of the possible embodiments were shown. Other aspects may, however, be embodied in many different forms and should not be construed as limited to the embodiments or examples set forth herein. Rather, these embodiments and examples were provided so that this disclosure was thorough and complete and fully conveyed the scope of the possible embodiments to those skilled in the art.

[0069] Although the embodiments have been described in language specific to structural features, methodological acts, and computer-readable media containing such acts, it is to be understood that the possible embodiments, as defined in the appended claims, are not necessarily limited to the specific structure, acts, or media described. One skilled in the art will recognize other embodiments or improvements that are within the scope and spirit of the present disclosure. Therefore, the specific structure, acts, or media are disclosed only as illustrative embodiments. The disclosure is defined by the appended claims.

CLAIMS:

What is claimed is:

1. A method of setting connection policies for a plurality of network paths from a first endpoint to a second endpoint, the method comprising:

5 receiving a first set of candidate addresses (112, 310) of the first endpoint (110) and a second set of candidate addresses (132, 320) of the second endpoint (130);

creating a plurality of candidate address pairs (142, 330), wherein each of the plurality of address pairs comprises an address of the first set of candidate addresses (112, 310) paired with an address of the second set of candidate addresses (132, 320);

10 determining a network path for each of the plurality of address pairs (142); and setting a connection policy for each of the plurality of network address pairs (142) and the associated network paths.

2. The method of claim 1, further comprising receiving session data, wherein the session data indicates the type of session to be established between the first endpoint and
15 the second endpoint.

3. The method of claim 2, wherein the session data includes a bandwidth request that indicates an amount of bandwidth requested for the type of session.

4. The method of claim 2, wherein the type of session is one of a voice-over internet protocol session, a video session, and a joint video voice-over internet session.

20 5. The method of claim 1, wherein the connection policy sets an amount of bandwidth available for each of the plurality of network address pairs.

6. The method of claim 1, further comprising, pre-failing one of the plurality of connection paths when the one of the plurality of network paths does not have available bandwidth to establish a session between the first endpoint and the second endpoint.

25 7. The method of claim 1, further comprising establishing a session between the first endpoint and the second endpoint using one of the plurality of network address pairs.

8. The method of claim 7, further comprising monitoring the established session to determine compliance with the connection policy associated with the session.

9. A computer storage medium encoding computer executable instructions that when
30 executed by a processor perform a method for setting connection policies for a plurality of network paths, the method comprising:

receiving a list of candidate addresses (112, 310) from a first endpoint (110) and a list of candidate addresses (132, 320) from a second endpoint (130);

combining one or more candidate addresses known by the server (140) with the list of candidate addresses (132, 320) from the second endpoint (130) to create a combined list of candidate addresses;

5 associating each address from the list of candidate addresses (112, 310) of the first endpoint (110) with each address from the combined list of candidate addresses to create a plurality of candidate address (142) pairs;

determining a network path for each of the plurality of candidate address pairs (142); and

10 setting a connection policy (152) for each of the plurality of candidate address pairs (142).

10. The method of claim 9, further comprising establishing a session between the first endpoint and the second endpoint, wherein the session is established using the network path of one of the plurality of candidate address pairs.

11. The method of claim 10, further comprising receiving session data prior to
15 establishing the session, wherein the session data indicates the type of session to be established between the first endpoint and the second endpoint.

12. The method of claim 10, further comprising receiving session data prior to establishing the session, wherein the session data indicates a desired amount of bandwidth for the session.

20 13. The method of claim 12, further comprising altering the desired amount of bandwidth for the session when it is determined that the connection policy for a selected one of the plurality of candidate address pairs does not have the desired amount of bandwidth available.

14. The method of claim 9 further comprising determining whether one or more of the
25 candidate addresses known by the server is behind a network address translator.

15. A system for setting connection policies for a plurality of network paths, the system comprising:

a first server device (140) configured to receive a first list of candidate addresses (112, 310) from a first endpoint (110) and a second list of candidate addresses (132, 320)
30 from a second endpoint (130), and wherein the first server (140) device is further configured to associate each candidate address (112, 310) from the first endpoint (110) with a candidate address (132, 320) from the second endpoint (130) to create a plurality of candidate address pairs (142), each of the plurality of network address pairs (142) having an associated network path; and

a policy server device (150) configured to set one or more policies (154) for each of the plurality of network address pairs (142).