



Avaya Aura®

White Paper for Dial Plan Processing by SIP Phones - Issue 1.0

Abstract

This document describes how dial plan patterns are generated by Session Manager's PPM service and used by the SIP phones. Improvements were recently introduced in both the PPM and the SIP phones. The description and rationale for these changes are described, as well potential problems which may be encountered in the field as a result of these changes.

Contents

1. Introduction.....	3
1.1. Confidentiality Statement.....	3
2. Background.....	3
3. Dial Plan Administration	4
3.1. Dial Plan Analysis.....	4
3.2. AAR Analysis	5
3.3. ARS Analysis	5
3.4. Location Specific Dial Plans	6
3.5. Emergency Numbers.....	6
4. Problems Necessitating Change.....	6
5. Changes to Dial Plan Algorithms	7
5.1. Higher Precedence to Specific Patterns (Endpoint change).....	7
5.2. No Pattern Absorption While Building Dial Plans (PPM change)	8
5.3. Smart Compaction (PPM change).....	8
5.4. Variable Length Pattern Enumeration (PPM change).....	8
6. Algorithms	9
6.1. Endpoint Algorithm.....	9
6.2. PPM Compaction Algorithm.....	10
6.2.1. Phase 1	10
6.2.2. Phase 2	11
7. Release Information	12
7.1. Session Manager	12
7.2. 96x0.....	12
7.3. 96x1	12
7.4. One-X Communicator	12
8. Impact to Customers	12
8.1. Calls Made Prematurely #1	12
8.2. Calls Made Prematurely #2	13
8.3. Calls Made Prematurely #3	13
8.4. Delay Calling Number	14
9. Compatibility Issues.....	15
9.1. Old 96x0/1 – New SM	15
9.2. New 96x0/1 – Old SM	16
9.3. New 96x0/1 – Old One-X Communicator	16
10. Conclusion	16

1. Introduction

Changes were recently made with how dial plan patterns are built by SM's PPM service and used by the SIP hard phones due to field issues with the prior implementation. So members of the hard phones (96x0 and 96x1), PPM and CM teams met and proposed changes for both PPM and the hard phones to address issues with the current implementation. These changes make dial plan processing consistent with CM's processing for non-SIP phones which is well proven. The changes have been incorporated into PPM and the phones. However, these changes may not be backwards compatible as customer's SMs and 96x0/1 phones are upgraded with these changes.

This white paper describes the creation of dial plan patterns in PPM, the processing of these patterns in the phones, the changes that were introduced, and the potential problems with remedies as these changes are deployed at customer sites. It should be noted that only SIP endpoints that use PPM's dial plan patterns are impacted. So 96x0, 96x1 and one-X Communicator are effected but Avaya Communicator and Summit are not.

1.1. Confidentiality Statement

The information disclosed herein is Avaya Confidential and Proprietary and is subject to the confidentiality provisions of our Master Services Agreement

2. Background

SIP telephones use dial plan patterns downloaded from PPM to determine when to send an INVITE to the CM. This eliminates the need for the user to explicitly tell the phone to make the call after entering the digits. Thus the behavior mimics that of DCP and H.323 phones where each digit entered by the user is processed by the CM and the CM makes the call when enough digits have been entered to match the CM dial plan. In the case of SIP and PPM, PPM translates the CM dial plans into list of patterns which are downloaded to the phones using the *getAllEndpointConfiguration* operation when the *DialPlanData* packet is requested.

The number of dial plan patterns that can be downloaded is 300. This limit is based on memory restrictions of the older SIP telephones. Although 300 sounds ample, many customers have huge dial plans that greatly exceed this limit. It's believed that these dial plans get large because new entries are added without cleanup of older, deprecated entries. Nonetheless when customers have large dial plans, PPM must reduce the number of patterns down to 300 through compaction.

Compaction of dial plan patterns involves replacing existing patterns with more general patterns containing wildcard characters. So the total number of patterns is reduced by introducing patterns that contain the 'x' and '+' wildcard characters. 'x' is any digit (i.e. 0-9) and '+' is 0 or more of any digit. So these patterns with wildcard characters replace more specific patterns.

When a user enters digits to make a call, the phone tries to match the digits with patterns downloaded from PPM. Sometimes there are multiple patterns which match the digits entered by a user. The desire is that as more digits are entered, more patterns are eliminated until only 1 pattern remains. When a digit is pressed resulting in a pattern to be fully matched, the phone

sends the INVITE. However, it's possible that there are multiple candidate patterns which are longer and so the phone can't send the INVITE. So it must wait in case the user wants to dial a number that matches a longer pattern. This is accomplished through an inter-digit timer and it's typically 5 seconds. So if the user's intent is to call a number that matches a shorter pattern amongst a set of matching patterns, the user must wait 5 seconds or press '#' which tells the phone to immediately send the INVITE.¹

Changes were recently made to the algorithm that the hard phones use to compare the collected digits against the PPM dial plan patterns. Also, changes were made to PPM's algorithm for building and compacting patterns to complement the endpoint changes. These are discussed in this white paper.

3. Dial Plan Administration

Dial plans are administered on CMs. PPM converts the dial plan entries into a list of patterns. When a user logs into their SIP phone, the phone will request the dial plan patterns using the *getAllEndpointConfiguration* request. PPM determines the user's CM and if applicable, the user's specific location administered on the CM. PPM then returns the list of patterns that correspond to the user's CM and location. Each pattern in the list may contain the following characters:

Characters	Translation
0 - 9	The literal digits that the user dials.
×	Any digit (i.e. 0 to 9).
+	A sequence of 0 or more digits.
Z	Play dial tone
*	Trunk access code delimiter used as the first character
#	Trunk access code delimiter used as the first character

PPM builds the dial plan patterns from the Dial Plan Analysis, AAR and ARS forms on the CM.

3.1. Dial Plan Analysis

For DPA, a pattern is built from each dialed string. If the total length exceeds the dialed string, then the pattern is padded with 'x' characters. So for example, suppose we have the following DPA entry:

Dialed String	Total Length	Call Type
313	7	ext

PPM creates the following pattern:

313xxxxx

¹ CM uses a 10 second inter-digit timeout. But if it matches a pattern and there is a longer candidate pattern, it will use a 3 second inter-digit in an attempt to use that longer pattern.

3.2. AAR Analysis

For AAR, PPM builds a pattern from each dialed string. It precedes the dialed string with the AAR access code followed by a 'Z'. Please note that in SM versions FP4 and earlier, PPM would generate a pattern even if there wasn't an access code configured. This pattern just contained the dialed string without the leading access code and 'Z'. Beginning in SM 7.0, there must be an access code or else PPM ignores the AAR entries.

As an example, suppose the AAR access code is '8' and we have the following AAR entry.

Dialed String	Total Min	Total Max	Call Type
2341	11	11	aar

PPM creates the following pattern:

8Z2341xxxxxxxx

If the total min and max lengths differ, PPM generates a pattern for each length. So with an AAR access code of '8' and the following entry:

Dialed String	Total Min	Total Max	Call Type
777	3	7	aar

PPM creates the following patterns:

8Z777
8Z777x
8Z777xx
8Z777xxx
8Z777xxxx

3.3. ARS Analysis

PPM generates patterns for ARS similarly to AAR. It builds a pattern using the ARS facility access code 1, followed by a 'Z', and then the dialed string. If there is an ARS facility access code 2, then a second pattern is generated. Beginning in SM 7.0, PPM will ignore ARS entries that don't have any administered access codes as is the case for AAR. Prior to SM 7.0, PPM would generate the patterns without the leading access codes and 'Z'.

As an example, suppose the ARS facility access code 1 is '9', the facility access code 2 is '*42' and we have the following ARS entry:

Dialed String	Total Min	Total Max	Call Type
15	11	11	fnpa

PPM creates the following patterns:

```
*42Z15xxxxxxxxxx  
9Z15xxxxxxxxxx
```

3.4. Location Specific Dial Plans

Dial plan entries in CM can be specified for *all* locations, or a specific location. When PPM starts up and also during the nightly audit, PPM will build and compact (if needed) the dial plan patterns for each location. It first builds the global dial plan patterns, referred to as “all” locations on the CM SAT. Then it builds the patterns for each CM location. These are built by creating patterns for each of the location-specific entries, and then merging in the global dial plan patterns. If the number of patterns exceeds 300 for any location, it will compact them down to 300.

When an endpoint requests dial plan data, PPM will determine the CM station associated with the user who’s logged onto the phone. Once the station is determined, PPM retrieves the CM location administered to that station. This administration is on the **CM Station** form via the **Location** field. PPM then uses the patterns that were built for that location. If no location is associated with a station, then PPM downloads the global dial plan patterns.

Note that for REMO users, the location is derived from the IP address of the SBC, mapped to a CM IP Network Map, which is associated with a CM IP Network Region. The Network Region has a location field that is used.

3.5. Emergency Numbers

Emergency numbers are merged into the dial plan patterns. PPM reserves 10 patterns per location for emergency numbers. So with the maximum number of dial plan patterns set to 300, PPM internally sets the limit to 290. So compaction occurs if the number of dial plan entries exceeds 290 for a given location, or the global patterns. Note that we still refer to this limit as 300 even in this document. But it’s actually 290 with the emergency number reservation.

When PPM downloads the dial plan patterns to an endpoint, it merges in any emergency numbers administered for the user’s SM location.

4. Problems Necessitating Change

A problem was encountered with a European customer who had a large dial plan containing variable length patterns which is typical with European dial plans. With the way PPM compacted the patterns and the way that the phones used them, the user was forced to wait 5 seconds before the call was made. Further analysis revealed that the dial plan processing algorithms that the phones used weren’t robust enough to provide a means to allow a shorter pattern to be used without delay if there were candidate longer patterns. This is unlike the CM algorithm used for non-SIP phones where the most precise pattern is given precedence. So CM provides a way for a pattern with fewer total digits to override a longer one by being more precise.

Members from the hard phones, PPM and CM met and decided that the phones should use the same algorithm that the CM uses when processing digits from non-SIP phones. This also meant that PPM needed to change the way it builds and compacts the dial plan patterns to ensure that shorter, more precise patterns are preserved.

The changes that were made to both the hard phones and PPM are described next.

5. Changes to Dial Plan Algorithms

Specifically, the changes are as follows:

1. Endpoints will give precedence to more specific patterns. The endpoint will use a pattern that has more digit characters over one that has less. They will also use a longer pattern over a shorter one that is terminated by a “+”.
2. PPM will no longer do pattern reductions when constructing the dial plan patterns. Before if PPM had patterns 611+ and 611555, PPM removed 611555 since it can be represented by 611+. PPM no longer does this during construction. Note it does do this during compaction.
3. If PPM has to compact the dial plan patterns because the total number exceeds the phone’s limits, it will reduce in such a way as to take advantage of the endpoint changes by minimizing the need for inter-digit timers when dialing numbers on the endpoint.
4. PPM now enumerates a variable length pattern to patterns representing each length within the length range of the pattern.

Details about each of the changes are provided below.

5.1. Higher Precedence to Specific Patterns (Endpoint change)

Suppose the endpoint had the following patterns:

```
6112xx  
611xxxx
```

If the user presses digits 6-1-1-2-3-4, the phone sends an INVITE even though there is a longer candidate pattern with 611xxxx. The reason is that the pattern 6112xx has 4 digits while 611xxxx had 3 digits. So 6112xx is more precise than 611xxxx.

The old endpoint would wait with an inter-digit timer after the ‘4’ is pressed since there is a longer matching pattern.

5.2. No Pattern Absorption While Building Dial Plans (PPM change)

Prior to these changes, PPM would automatically absorb more precise patterns into patterns with wildcard characters. Suppose PPM had the following patterns:

```
611+
611103xxx
611104xxx
61112xxxx
61119xxxx
611555
61182xxx
61190xxx
61191xxx
61192xxx
```

PPM would just download 611+ since the remaining patterns are absorbed by this pattern. But with endpoints selecting the more precise pattern, PPM keeps the patterns beginning with 611 as the endpoints will now prefer those over 611+.

Note that the old endpoints would only use 611+ and ignore the remaining patterns. Any number beginning with 611 will have a delay of 5 seconds before being called.

5.3. Smart Compaction (PPM change)

PPM will avoid compactions that will introduce an inter-digit timer for numbers that didn't require it before compaction. Suppose PPM has the following patterns and needs to compact:

```
6112xx
6113xx
61141x
6114xxx
```

PPM can only compact to the following:

```
611xxx
61141x
6114xxx
```

PPM won't reduce 61141x with 611xxx or else there'd be no way to dial a number such as 611412 without an inter-digit delay as the phone would apply the longer 6114xxx pattern.

5.4. Variable Length Pattern Enumeration (PPM change)

PPM now enumerates a variable length pattern. Suppose on the CM we have a pattern beginning with digits 833 whose min length is 5 and max length is 9. PPM used to generate the following:

833xx+

With this pattern, the phone must use a timer before calling any number of length 5 or more beginning with 833.

But now PPM generates the following:

833xx
833xxx
833xxxx
833xxxxx
833xxxxxx

The advantage to this change is that the phone can call the number immediately if the digits match the last (longest) pattern. So a number of length 9 beginning with digits 833 can be dialed immediately. This is an improvement over the previous case where a timer was required for any number of length 5 or more beginning with digits 833..

6. Algorithms

The algorithm used by the endpoints when processing the entered digits with the dialplan patterns is important to understand. If the phone's dialing behavior is unexpected, knowing the algorithm will allow an administrator to fix the unexpected behavior by changing the CM dial plan administration.

PPM's algorithm for building and compacting patterns is important to understand if inter-digit timers are used by the endpoints when dialing certain numbers. PPM is deployed as a service under SM's JBoss container. When it initializes, it constructs the dial plan patterns for each CM location. These pre-built lists of patterns are downloaded to phones upon request. If the dial plan is changed on a CM, then PPM will rebuild the dial plan patterns for the effected CM. Then the phones for users administered to the CM are notified to reload their configuration, which includes the dial plan patterns. The amount of time for this can vary but should typically be less than 5 minutes from when the SM database is updated with the CM dial plan change.

6.1. Endpoint Algorithm

As digits are entered, the endpoint maintains a set of candidate patterns that match the digits. Note some patterns may match directly on the digit and others using a wildcard character of 'x'. As more digits are entered, some candidate patterns are eliminated if an entered digit doesn't match the next character (and it's not a wildcard character) in these candidate patterns. If a pattern is completely matched, call the number if the remaining longer candidate patterns are less precise (have fewer actual digits). This means that the SIP phone will continue to collect digits if the longer candidate patterns have the same or more precision than the currently matched pattern(s).

If a '+' character is encountered in a candidate pattern, it's used only if there are no other longer candidate patterns. Note that patterns terminated with a '+' have an unknown maximum length, forcing the endpoint to use an inter-digit time-out before calling the number. If the user presses the '#' button, the digits that are entered are dialed immediately. If no patterns are fully matched, then the phone calls the number after the inter-digit timer expires. For this case, it's important to note that the user dials a number that is outside the dial plan and the phone will make the call after the timer expires. **This introduces potential toll fraud which must be trapped by the CM or SM.**

There are 2 key differences with this new algorithm over the old:

1. Patterns with more precision have higher precedence over less precise patterns. The old phones always try to match the longest pattern, regardless of precision.
2. If the old algorithm encountered a '+' in a pattern while trying to match an entered digit, that pattern was selected and all other patterns were ignored. This forced the use of an inter-digit time-out before making the call. Now this is no longer the case.

6.2. PPM Compaction Algorithm

If the total number of patterns built for a CM location exceeds 300, PPM will compact those patterns. This is done in 2 phases. The first phase is conservative and the second phase is more aggressive. If the first phase does not sufficiently reduce the number of patterns, then the second is performed. The phases are below:

1. Reduce existing patterns by replacing them with generated patterns incorporating 'x' characters that don't require inter-digit timers (i.e. can be dialed immediately).
2. Reduce existing patterns using generated patterns which are terminated with a '+'. Patterns that already require a timer are reduced first so as to not negatively impact the user experience by introducing delays when calling certain numbers. If more compaction is required, then patterns that on their own don't need a timer are reduced. This will introduce delays when calling certain numbers.

6.2.1. Phase 1

In this phase, PPM targets patterns of the same length that don't require inter-digit timer with a pattern of the same length that uses the 'x' wildcard character. As an example, suppose we have the following patterns:

```
3035381xxx  
3035382xxx  
3035383xxx  
3035384xxx  
3035385xxx  
3035386xxx
```

PPM will reduce these 6 patterns to a single pattern:

303538xxxx

6.2.2. Phase 2

For this phase, PPM will compact using patterns that are terminated with the '+' wildcard character. Targeted first are patterns that already require an inter-digit timer. If more compaction is needed, then patterns that don't need a timer are reduced. This is the case where the user experience will be negatively impacted.

An enumeration of a variable length pattern illustrates both aspects of this phase. Suppose we have a dial plan entry beginning with 833 with a min length of 5 and a max length of 9. PPM generates the following:

833xx
833xxx
833xxxx
833xxxxx
833xxxxxx

If compaction is performed and PPM targets these patterns, PPM will reduce the patterns already requiring inter-digit timers (which are the first 4 pattern) and leave the pattern that can be dialed immediately intact by reducing this set of patterns to:

833xx+
833xxxxxx

The above patterns don't impact the user experience at all. A 9 digit number that begins with 833 is still dialed immediately. Anything shorter requires a timer which was the case even before the reduction was performed.

If after doing the above reduction, the number of patterns still exceeds 300, then PPM must reduce patterns that can be dialed immediately. So now the enumeration will be reduced to just a single pattern:

833xx+

Now if the user dials a 9 digit number beginning with 833, there will be a 5 second delay before the phone makes the call. So the user experience is now negatively impacted. It is important to note that this is only done as a last resort.

The key differences with the new algorithm over the old are:

1. PPM no longer does pattern absorption when simply building (not compacting) the dial plan. This is so that we don't remove patterns with more precision over less precise patterns that use wildcard characters.

2. PPM is now keenly aware of which patterns require timers and which ones don't. So during compaction, it will reduce patterns in such a way as to not introduce inter-digit timer for numbers that didn't need them previously, except as a last resort.

7. Release Information

The releases of SM and the endpoints for which these changes were introduced are described here. Ideally, the SM is upgraded prior to the endpoints to avoid potential issues. Please see section 9 for more details.

7.1. Session Manager

The SM changes were added in 6.3.12. It should be noted that problems were encountered in the field with the PPM's compaction algorithm which required a fix that went into 6.3.14. Specifically, PPM could compact in such a way as to generate patterns with a leading 'x' which caused problems in the endpoints. So as of 6.3.14, PPM will no longer generate patterns that begin with an 'x'.

7.2. 96x0

The 96x0 changes went into release 2.6.13.

7.3. 96x1

The 96x1 changes went into release 6.5.0.

7.4. One-X Communicator

One-X Communicator is in the process of changing its dial-plan processing algorithm to be consistent with the 96x0 and 96x1 phones. These changes are planned for R6.2-SP7.

8. Impact to Customers

These changes may introduce dialing behavior changes to existing customers. The typical problem will be that a number is dialed before all of the digits have been entered. The remediation of this negative behavior requires changes to the customers dial plans administered on the CM. Examples of this following in the subsequent subsections.

8.1. Calls Made Prematurely #1

After upgrading the SM and endpoints, calls are made before all of the digits have been pressed. This is due to the endpoint choosing a shorter, more precise pattern over a longer pattern. If we have the following patterns:

```
3134x  
313xxxxxxxx
```

If the user dials 3134872299, the old endpoint applies the second pattern as it's longer and makes the call after the last '9' is pressed. The new endpoint will apply the first pattern and call 31348

after the '8' is pressed. To remedy this, we need to add a longer more precise pattern. So adding a CM DPA entry with leading digits 31348 and length 10 causes PPM to now send:

```
3134x
31348xxxxxx
313xxxxxxx
```

The new endpoint will now prefer the second pattern with 5 digits over the first which has 4 digits.

8.2. Calls Made Prematurely #2

In this example, the user attempts to dial a number beginning with 111 and the phone calls 111, disregarding any remaining digits. PPM sends the following dial plan patterns:

```
11x
120
1x
1xx
1xxx
1xxxx
1xxxxxxxxxx
1xxxxxxxxxx
1xxxxxxxxxx
1xxxxxxxxxx
```

If the user wants to dial 1114567892233 which corresponds to the last pattern, the old endpoint uses the longest pattern which is 1xxxxxxxxxx. However, the new endpoint calls 111 after the last '1' is pressed. The reason is that pattern 11x (with 2 digits) is more precise over 1xxxxxxxxxx (with 1 digit). This is best remedied by deleting 11x from the CM dialplan (e.g. begins with 11 and has length 3).

8.3. Calls Made Prematurely #3

This rarer case can happen if PPM generates a shorter pattern terminated with a '+' and there are longer patterns with the same leading digits as the shorter pattern. Suppose PPM sends the following:

```
6112x
611+
```

The old phone will always choose the second pattern because it gave precedence to patterns with a '+' over other patterns. So the user could dial 6112345 and the phone would make the call after the inter-digit timer expires. But the new phone will call 61123 after the '3' is pressed.

To remedy this, the pattern that generates the 6112x should be removed, leaving just 611+.

8.4. Delay Calling Number

A customer recently experienced an inter-digit timeout when dialing 911 after the SM was upgraded to 6.3.12 but didn't upgrade the 96x1 phones to 6.5. The 96x1 6.5 FW did not suffer from the 5 second delay like the older FW. Investigation revealed that this delay occurred prior to the SM upgrade. So it wasn't due to the SM upgrade after all and was probably due to a CM dial plan administration change. The customer had over 33k 96x1 phones which made upgrading to 6.5 a non-viable solution. So changes were made to the CM dialplan to ensure that there was no ambiguity when the user pressed 9-1-1 and would dial it immediately.

The delay resulted from the following DPA pattern entries associated with location "all":

```
911
9xxxxxxxxx
```

When the user dials 9-1-1, the old 96x1 FW uses the second pattern because it gives precedence to the longest pattern. The new FW uses the first pattern because it's more precise than the second pattern (3 digits versus 1). This is a prime example of why this change to the endpoints dialplan processing was needed!

Since the customer couldn't upgrade the 96x1 phones as a quick solution, the CM dialplan was changed to accommodate the old 96x1 FW. The 9xxxxxxxxx pattern was removed and replaced with the patterns that didn't provide an alternative when dialing 9-1-1. In addition, bogus patterns were added to ensure that PPM didn't compact the new patterns back to 9xxxxxxxxx. So the changes to the global DPA were the following:

Action	Dialed String	Total Length	PPM Pattern
Delete	9	10	9xxxxxxxxx
Add	90	10	90xxxxxxxx
Add	910	9	910xxxxxx
Add	910	10	910xxxxxxxx
Add	912	9	912xxxxxx
Add	912	10	912xxxxxxxx
Add	913	9	913xxxxxx
Add	913	10	913xxxxxxxx
Add	914	9	914xxxxxx
Add	914	10	914xxxxxxxx
Add	915	9	915xxxxxx
Add	915	10	915xxxxxxxx
Add	916	9	916xxxxxx
Add	916	10	916xxxxxxxx
Add	917	9	917xxxxxx
Add	917	10	917xxxxxxxx
Add	918	9	918xxxxxx

Add	918	10	918xxxxxxxx
Add	919	9	919xxxxxxx
Add	919	10	919xxxxxxxx
Add	92	9	92xxxxxxx
Add	92	10	92xxxxxxxx
Add	93	9	93xxxxxxx
Add	93	10	93xxxxxxxx
Add	94	9	94xxxxxxx
Add	94	10	94xxxxxxxx
Add	95	9	95xxxxxxx
Add	95	10	95xxxxxxxx
Add	96	9	96xxxxxxx
Add	96	10	96xxxxxxxx
Add	97	9	97xxxxxxx
Add	97	10	97xxxxxxxx
Add	98	9	98xxxxxxx
Add	98	10	98xxxxxxxx
Add	99	9	99xxxxxxx
Add	99	10	99xxxxxxxx

The bogus patterns are the ones that have total length of 9. Without these, PPM would compact the patterns back down to 9xxxxxxxxx. PPM only had to use the phase 1 compaction where it merged patterns of the same length, without introducing any patterns terminated with a '+' which is phase 2 compaction. So the PPM patterns were downloaded verbatim.

These additional entries added for the old 96x1 FW didn't adversely impact the new 96x1 FW. It still calls 911 immediately even with all of these added entries. This example clearly illustrates the motivation for changing the dialplan SW in both the endpoints and PPM.

9. Compatibility Issues

In this section we discuss possible compatibility issues involved between phones and SMs. For example, what happens if I upgrade my SM but not the endpoints? Or I upgrade the endpoints but not the SM?

9.1. Old 96x0/1 – New SM

For this case, the SM has been upgraded to 6.3.12 or newer but the endpoints are not upgraded. The only impact is actually a positive one where inter-digit timer may no longer be needed for a variable length pattern. The old PPM simply generated a single pattern representing the shortest length terminated with a '+'. So the phone would always use an inter-digit timer when calling numbers for this variable length pattern because it didn't know the maximum length.

But the new PPM enumerates this pattern into patterns for each distinct length. So the old phone will use the longest pattern which is a fixed length and thus doesn't require an inter-digit timer to call.

9.2. New 96x0/1 – Old SM

For this, we may see cases where the endpoint calls a number prematurely whereas the old phone did not. For this, the phone is using a shorter more precise pattern over a longer one with less precision. The shorter pattern is probably no longer used and should be removed from the CM's dial plan administration.

9.3. New 96x0/1 – Old One-X Communicator

If a user has both a new hard phone and an old One-X Communicator, that user may see cases where the hard-phone will initiate a call immediately where the One-X Communicator in off-hook mode will wait 5 seconds before calling the number. Since the soft phone uses the old dial plan processing algorithm, this can occur. This remedied by upgrading to a new version of one-X Communicator. Or else live with the 5 second delay on the soft phone, or enter a '#' at the end of the number.

This case is rare since it only impacts users who place calls using one-X Communicator by going off-hook first and then dialing. The more common usage of one-X Communicator is to enter the entire number first before initiating the call.

10. Conclusion

The changes made to both the endpoints and PPM for dial plan processing will allow for an improved user-experience by reducing the need for inter-digit timers when making calls. With knowledge of how these phones process the dial plan patterns, administrators can update the CM dial plans to take full advantage of these changes.

©2015 Avaya Inc. All Rights Reserved.

Avaya and the Avaya Logo are trademarks of Avaya Inc. All trademarks identified by ® and ™ are registered trademarks or trademarks, respectively, of Avaya Inc. All other trademarks are the property of their respective owners. The information provided in these Application Notes is subject to change without notice. The configurations, technical data, and recommendations provided in these Application Notes are believed to be accurate and dependable, but are presented without express or implied warranty. Users are responsible for their application of any products specified in these Application Notes.

Please e-mail any questions or comments pertaining to these Application Notes along with the full title name and filename, located in the lower right corner, directly to the Avaya DevConnect Program at devconnect@avaya.com.